
théoric
LA REVUE DES PASSIONNES D'ORIC
ISSN 0762-6711
• UTILITEL :
POUR UTILISER LE MINITEL
• PUISSANCE 4 : CONTRE ORIC
• MIXTE :
PARTAGEZ L'ÉCRAN
• CLASSEMENT INSTANTANÉ :
COMMENT TRIER
M2635-20-30F
N° 20 - MENSUEL 30 F
MAI-JUIN 1986

EDITORIAL



Vous avez vu la couverture ? THEORIC n° 20 déjà ! Que de chemin parcouru depuis notre numéro 1. Certains donnaient l'ORIC pour mort et tout semble démontrer le contraire :

— un pavé numérique et commercialisé depuis peu par la société ICV : c'est le CALCULATOR,

— les premières commandes de TELESTRAT sont enregistrées par EUREKA : l'ordinateur épouse la télématique,

— quant à nous, SORACOM, nous continuons à marquer notre confiance envers ce produit en sortant le livre "L'ORIC A NU" que tout programmeur averti doit posséder.

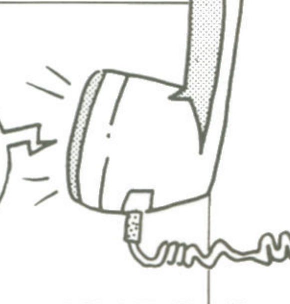
Et vous, les utilisateurs ? Ça va bien, merci ! La preuve ! Tous les bons articles et programmes qui nous parviennent à la rédaction. De quoi alimenter les prochains numéros...

théoric n°20 COUPON A JOINDRE A TOUTE CORRESPONDANCE

Une ligne téléphonique est à votre disposition, vous mettant en contact direct avec la rédaction. Ceci est un service sans égal ! Respectez simplement les horaires et les jours que nous vous indiquons :
MERCREDI de 9 h à 12 h et de 14 h à 17 h.
VENDREDI de 9 h à 12 h seulement.
Tout appel en dehors de ces créneaux sera refoulé : ne dépensez pas inutilement votre argent !

Le numéro : **99.52.98.11.**





SOMMAIRE

THEORIC
Publication mensuelle
Editions SORACOM
La Haie du Pan - 35170 BRUZ
Tél.: 99.52.98.11
lignes groupées
Télex : 741 042 F
Code APE
5120
Directeur de publication
Sylvio FAUREZ
Rédacteur en chef
Denis BONOMO
Secrétaire de rédaction
Florence MELLET
Maquette
Jean-Luc AULNETTE
Patricia MANGIN
Abonnements, Ventes, Réassorts
Catherine FAUREZ
Composition
FIDELTEX
Impression
VAN DEN BRUGGE
Distribution
NMPP
Publicité
IZARD CREATIONS
Tél.: 99.31.64.73.

THEORIC est un mensuel édité par la Sarl SORACOM, expirant le 22 septembre 2079, au capital de 50 000 francs. S. FAUREZ en est le gérant, représentant légal. L'actionnaire majoritaire est Florence MELLET.
Dépôt légal à parution



I.C.V

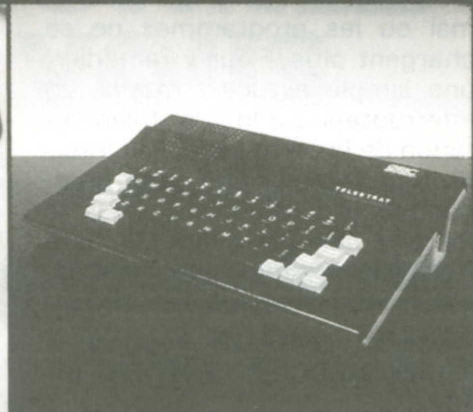
REVENDEUR AGRÉE ORIC EUREKA

ouvert du mardi au samedi de 9 h 30 à 12 h 15 et de 14 h 30 à 19 h 15
130, ROUTE DE CORBEIL — 91360 VILLEMOISSON-SUR-ORGE
Tél. (1) 69.04.04.50

I.C.V. A L'AVANT-GARDE DES NOUVEAUTÉS







LECTEUR DISQUETTES 3,5 p

- Lecteur esclave 3"1/2
- Double têtes
- Double densité
- Alimentation 220 volts
- Cordon de raccord
- Compatible MICRODISC/JASMIN 2

1990 F

PAVÉ NUMÉRIQUE/ JOYSTICK

- 14 touches (0 à 9), le point, DEC, FONCTION, ENTER.
- 1 prise joystick (fonctionne sur les 4 flèches et la barre d'espacement), compatibilité assurée sur 75% des jeux.

490 F

TELESTRAT

- 64 Ko de RAM (extensible)
- 48 ko de ROM (extensible)
- CARTOUCHE HYPER BASIC
- CARTOUCHE TÉLÉMATIC
- Lecteur disquette 3 pouces
- Câbles, manuels

3990 F

NOUVEAUTÉS DISPONIBLES ET EN DÉMONSTRATION

- TELESTRAT 64 K .... 3990 F
- ATMOS 48 K .... 990 F
- Moniteur couleur .... 2750 F
- Moniteur monochrome vert HR12 1150 F
- Imprimante MCP40 plotter 4 couleurs .... 990 F
- Imprimante BROTHER 1009 ..... 1995 F
- Option picots BROTHER 1009 .. 350 F
- Lecteur disquettes SEDORIC (maître) .... 2490 F
- Lecteur disquettes (auxiliaire) .... 1490 F

ACCESSOIRES

- Modulateur noir et blanc UHF .. 295 F
- Modulateur couleur UHF .... 495 F
- Câble pour imprimante parallèle 150 F
- Rouleau papier MCP 40 .... 18 F
- Rame papier imp. 80 (SOOF) ... 79 F
- Bloc Oric 9 volts .... 150 F
- Joystick type "Quickshot 1" ..... 95 F
- Disquette 3,5 pouces (MAXELL) .. 25 F

Revendeurs, nous consulter

LOGICIELS

NOUVEAU : T.B.D (5 utilitaires sur disquettes Sedovic) .... 250 F
T.B.D - Transfert K7 vers disquettes
CAT - Transfert TDOS vers Sedoric
DTR - Dump mémoire en temps réel
DD - Test piste défectueuse
RTP - Répare une piste défectueuse

+ de 70 titres en magasin

ICV distributeur de la revue THEORIC

BON DE COMMANDE

I.C.V. 130, Route de Corbeil — 91360 VILLEMOISSON-SUR-ORGE

NOM : ....
PRÉNOM : ....
ADRESSE : ....
Code Postal : ....
VILLE : ....
TÉL : ....

DÉSIGNATION	Qté	P.U	P.T
....	....	....	....
....	....	....	....
....	....	....	....
....	....	....	....
....	....	....	....

Frais de Port
- 30 F pour achats inférieurs à 500 F
- 50 F pour achats de 500 F à 1000F
- 70 F pour achats supérieur à 1000 F

Ci-joint un chèque de F. : ____________

Frais de Port : ________

TOTAL ________

IZARD Créations - 99 31 64 73



Jean-Louis JAROUSSE

Le bus de l'ORIC est déjà bien chargé lorsqu'on le connecte au Jasmin ou au Microdisc. Le relier, en plus, à la carte série décrite dans THEORIC peut poser des problèmes : le système s'initialise mal ou les programmes ne se chargent plus. Pour y remédier, une simple astuce : mettre un interrupteur sur le +5 V alimentation de la carte série. Cet interrupteur sera sur OFF pour charger les programmes ; vous le mettrez sur ON avant de faire RUN. C'est simple et efficace.

Frédéric TARAUD répond à Patrick MERCIER (THEORIC n° 16)

Pour faire fonctionner la carte série dans Micro et Robots, il suffit de relier par un petit pont de soudure les pattes 1 et 2 du 7427.
Cette liaison, figurant sur le schéma de principe, a été oubliée sur le circuit imprimé.

Alain FERNANDEZ
30 CHUSCLAN

Un ami m'a prêté un drive APPLE. Puis-je le connecter au contrôleur Jasmin de l'ORIC ?
Le JASMIN peut être utilisé avec des drives 5''1/4 à condition que ceux-ci soient à la norme Shugart. Cette norme définit des temps d'accès, vitesse de rotation, etc. ainsi qu'un certain ordre pour le brochage des connecteurs. Si votre drive répond à cette norme, il pourra être piloté par le contrôleur Jasmin. Hélas, dans les drives ''compatible APPLE'', c'est peu souvent le cas !

Patrick CHANTEREL
92 VANVES

A quand le Pascal sur ORIC ?
Pas de réponse à cette question... Nous doutons que ce langage soit un jour implanté sur notre machine favorite, vu la désaffection des principaux gros éditeurs de logiciels pour ORIC.

clubs

BERGERAC compte un club de micro-informatique (en dehors des écoles) qui fonctionne selon la loi de 1901 sur les associations :
Son nom : MICRO 24.
Adresse du siège : 6, Allée André Levèque, 24100 BERGERAC.
Adresse des réunions : 18 rue de la Résistance, 24100 BERGERAC (tous les jeudis à partir de 20h30).
Matériels utilisés : AMSTRAD (464 et 6128), ORIC (ORIC-1 et ATMOS), APPLE (APPLE IIc et MACINTOSH), THOMSON (MO5).
But : réunir tous les fervents de la micro-informatique afin de partager des idées, de développer en commun, etc. Les mordus qui ne se connaissent pas encore sont attendus afin de leur inoculer le virus.
Président : T. BRUNET.

CALCULATOR: Essai du pavé numérique



Nous vous présentions en exclusivité, dès sa sortie sur le marché, le pavé numérique "CALCULATOR" conçu et commercialisé par la société ICV. Depuis, nous avons eu l'occasion d'en effectuer l'essai : voici nos impressions.

CALCULATOR est livré avec une petite feuille "notice" destinée à faciliter son branchement sur l'ORIC. Rappelons, pour ceux qui n'auraient pas lu notre précédent article, que CALCULATOR est un pavé numérique proposant, en plus, une touche "FUNCTION" qui sera bien vue par tous les possesseurs d'ORIC-1 et une prise JOYSTICK. L'idée est bonne, et tous ceux qui utilisent habituellement leur matériel pour saisir de longues séries de nombres (que ce soient des DATA ou des valeurs numériques de leur budget prévisionnel) trouveront bien pratique cette nouvelle "extension".

CALCULATOR n'a immobilisé ni le bus d'extension, ni la prise imprimante, pour le connecter à l'ORIÇ, il suffit d'ouvrir la machine et de brancher, sur le clavier, le connecteur situé au bout du câble du CALCULATOR. En règle générale, cela ne posera pas de probème, mais, néanmoins, tout a été prévu pour les autres cas.

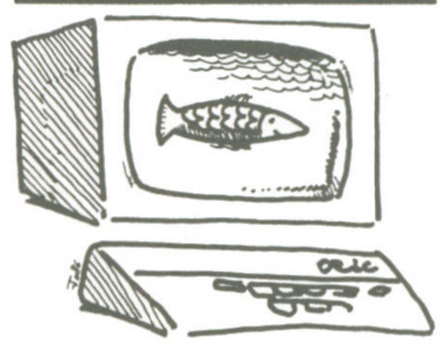
Sur notre ATMOS, la greffe a été aisée : après avoir ouvert la machine, il suffit d'enfiler le connecteur sur celui du clavier. Sur certains modèles d'ATMOS ou d'ORIC-1, il faudra changer ce connecteur clavier. Cette éventualité a été prévue et un connecteur est livré avec le CALCULATOR. Cela implique de savoir tenir et utiliser proprement un fer à souder. Si ce n'est pas votre cas, contactez votre revendeur qui effectuera, ou fera effectuer par un technicien, la petite transformation.

Le câble ombilical qui relie le CALCULATOR à l'ORIC passe par la découpe du connecteur imprimante. Nos craintes ont été vite dissipées : cela n'empêche pas de brancher l'imprimante. Quant à l'adaptation sur le TELESTRAT, pas de problème, elle a été prévue !

L'utilisation du CALCULATOR est bien agréable, surtout lorsqu'on est habitué à travailler sur un clavier de machine à calculer ; la saisie de listes de nombres n'est plus une corvée. Si, de plus, vous avez la chance de posséder un MICRODISC et son SEDORIC, vous découvrirez alors toute la puissance de CALCULATOR, grâce à la pleine utilisation de la touche FUNCTION !

Dernier point : le prix. Vu l'intérêt porté par les utilisateurs d'ORIC au CALCULATOR, la société ICV a décidé de baisser son prix de 590 à 490 F. Quand on sait que le CALCULATOR est équipé de touches professionnelles, à contacts dorés, il semble difficile de faire mieux, surtout si l'on considère la présence de la prise joystick !

Maintenant, qu'on l'a, on le garde !



ECRAN PLAT SUR ORIC

L'idée était séduisante, n'est-ce pas ? Elle n'aura pas vécu longtemps, et c'est dommage ! Il est vrai que, à 700 F, la société SOLE aurait eu bien du mal à fabriquer cet écran.

Toutes nos excuses aux lecteurs qui ont mordu à l'hameçon... Mois d'avril oblige... Le canulard était gros et pourtant !



Dans l'excellent programme "LEONARD" s'est glissée une erreur vicieuse... Prenez votre listing (programme "MENU") et ajoutez au stylo rouge une ligne 435 L=L+10.

Maintenant, l'ordinateur ne vous sortira plus bêtement "Erreur en ligne 10000" mais bien sûr la ligne concernée.

Merci Philippe LAFORGUE !

ABONNEZ-VOUS.



théoric

VALABLE POUR 11 NUMEROS
(attention : juillet et août un seul numéro)

20

Ci-joint un chèque de . . . . . . F à l'ordre de SORACOM,
Editions SORACOM, La Haie de Pan
35170 BRUZ

NOM . . . . Prénom . . . .
Adresse . . . .
Code Postal . . . . Ville . . . .
Pays . . . . Signature . . . .

Abonnement 3 mois (une seule fois) . . . . 85 F
Abonnement 6 mois . . . . 160 F
Abonnement 1 an France métropolitaine . . . . 270 F
Tarif avion . . . . 270 + 140 F

SORACOM éditions

LES DISQUETTES DE THEORIC

Tous les programmes de **THEORIC,** groupés par deux numéros sur une disquette. Plus de perte de temps pour les taper... Existe pour le moment en **JASMIN** seulement.

Le prix : **135 F** pour les abonnés (joindre impérativement l'étiquette), **165 F** pour les autres.
Avertissement : les programmes stockés sur ces disquettes sont tels que parus dans **THEORIC.** A vous de les adapter à vos besoins et au fonctionnement sur disque lorsque nécessaire.

N° 1 THEORIC n° 4 et 5
N° 2 THEORIC n° 6 et 7
N° 3 THEORIC n° 8 et 9
N° 4 THEORIC n° 10 et 12
N° 5 THEORIC n° 13 et 14
N° 6 THEORIC n° 15 et 16
N° 7 THEORIC n° 17 et 18
N° 8 THEORIC n° 19 et 20
N° HS THEORIC HORS SERIE (n° 11)

```
VOLUME : TO19&20

U LEORIC1 .BAS  S    74 SECTORS
U EUROPE  .BAS  S    52 SECTORS
U BULLELM .BAS  S     6 SECTORS
U MIROIR  .BAS  S    14 SECTORS
U DEUXSPHE.BAS  S     9 SECTORS
U RWSED   .BAS  S     4 SECTORS
U DUMP    .BAS  S     7 SECTORS
U CHECSEC .BAS  S     5 SECTORS
U COMPTE  .BAS  S    18 SECTORS
U FENETRE .BIN  S    18 SECTORS
U UTIL4   .BAS  S    14 SECTORS
U LOGO    .BAS  S    14 SECTORS
U MININTER.BAS  S     7 SECTORS
U CODASC  .BAS  S     6 SECTORS
U TABLEAUX.BAS  S     6 SECTORS
U INTERPRE.BAS  S    35 SECTORS
U MYSTERE .BAS  S    16 SECTORS
U PUISSAN4.BAS  S    55 SECTORS
U ECRAMIXT.BAS  S     8 SECTORS
U FASTEX80.BIN  S     3 SECTORS
U UTILIDAT.BAS  S    10 SECTORS
U UTILITEL.BIN  S     3 SECTORS
U UTILITEL.BAS  S    20 SECTORS
U VALOCAT .BAS  S    14 SECTORS
U VALOCT01.BAS  S     8 SECTORS
U MENULE01.BIN  S    10 SECTORS
  259 SECTORS FREE
```

BON DE COMMANDE DES DISQUETTES THEORIC

TO n°20

Je commande les disquettes 1 2 3 4 5 HS 6 7 8
Ci-joint : 135 F par disquette et mon étiquette abonné
165 F par disquette car je ne suis pas abonné

Soit au total la somme de . . . . . . F réglée par chèque ci-joint.

NOM . . . . Prénom . . . .
Adresse . . . .
Code Postal . . . . Ville . . . .

ECRAN MIXTE

Thierry LEGAL

Sur ORIC, l'utilisateur a le choix entre deux modes d'écran : le mode TEXT ou LORES qui permet d'utiliser 28 lignes de 40 caractères, et le mode haute résolution (HIRES) qui donne 200 lignes graphiques de 240 pixels × 3 lignes TEXT.

Pour cela il suffit de tromper l'ORIC-1 (ou l'ATMOS) en jouant sur la synchronisation. Vous avez déjà, sans doute, remarqué que les combinaisons "ESC" + une touche parmi "X, Y, Z |, /, | et DEL" provoquaient parfois des perturbations à l'affichage. Il y a deux explications à cela : la première est qu'en France on utilise du 50 Hz et que certains de ces codes font passer en 60 Hz (norme anglaise) provoquant ainsi une perte de la synchronisation du balayage (essayez "ESC" + "Z", par exemple, puis "ESC" + "X" pour revenir à la normale). La seconde explication provient du fait que les autres codes servent à définir si on est en TEXT ou en HIRES et donc à indiquer si les octets doivent être compris comme des caractères ou comme du graphisme.
C'est le dernier de ces codes que nous allons utiliser. Tapez, par exemple, la séquence suivante :
TEXT:PAPER 0:INK 6:HIRES:
PAPER 4:DRAW 239,134,1:
TEXT
Tapez ensuite : ESC puis DEL. On voit alors apparaître une partie de l'écran HIRES sur l'écran TEXT ! C'est ce phénomène que nous allons exploiter.
Il nous faut pour cela stabiliser cet écran. Cela sera fait tant que le code ESC + DEL restera affiché à l'écran. En effet, appuyer sur la flèche ↓ (curseur bas) : au bout d'un moment, l'écran revient en mode TEXT car le scolling a fait "sortir" le code de l'écran.
Nous allons donc condamner la dernière ligne (26) où l'on placera ce code. A ce propos, il y a plusieurs façons de l'y mettre :
— ESC puis DEL en mode direct sur la ligne 26.
— ? @ 0,26;CHR(27);CHR(127); qui peut aussi s'écrire ? @ 0,26; CHR(159);
— PLOT 0,26,31
C'est la dernière solution (la plus simple) que l'on retiendra. Il ne reste plus qu'à positionner correctement les pointeurs définissant la "page TEXT" pour que la dernière ligne ne puisse plus être modifiée.
Les modifications sont ici différentes sur ORIC-1 et sur ATMOS. Cela provient du fait que la routine de défilement de l'écran (scrolling) a été revue sur ce dernier. Afin d'accélérer ce scrolling, certaines valeurs sont calculées une fois pour toutes, et, la méthode d'adressage a été améliorée en utilisant une adresse intermédiaire de plus. Le gain de temps réalisé est de l'ordre de 30 %. Mais, revenons au problème.
Sur ATMOS
1) DOKE 632,48720 (#BE60) adresse seconde ligne.
2) DOKE 634,48680 (#BE28) adresse première ligne.
On définit ainsi la 16e ligne habituelle comme étant maintenant la première. Il reste donc 11 lignes, mais on a choisi de ne pas toucher à la dernière. On dispose donc de 10 lignes vraiment utilisables. Lors du scrolling, on aura donc 40 (10 − 1) = 360 octets à déplacer. Cela donne :
3) DOKE 636,360 et
4) POKE 638,10
Sur ORIC-1
1) DOKE 621,48640 (#BE00) adresse première ligne
2) POKE 623,10 Nombre de lignes utilisables.
Après avoir effectué ces opérations, faites :
CLS:FOR I = 1 TO 100:? I:NEXT
et voyez le résultat (création de la fenêtre).
REMARQUE : Cette procédure de définition de la "page Text" est tout à fait "standard". Ainsi, si on désire normalement 28 lignes au lieu de 27, il suffit de faire :
DOKE 632, #BBA8:DOKE 634: #BB80:DOKE 636,1080:POKE 638,28:CLS si on a un ATMOS et, si on a un ORIC-1 :
DOKE 621,#BB58 (47960):
POKE 623,10:CLS
Cela peut parfois être intéressant.
En reprenant notre écran "mixte", il est intéressant d'étudier sa disposition en mémoire. Les résultats sont portés sur le tableau ci-joint. On y observe :
— HIRES de #A000 à #B3FF et même #B4FF car page #B4 vide.
— TEXT de #BE28 à #BFE0
— Reste de #B800 à #BE27 qui est disponible pour des routines en langage machine, par exemple.
Le Basic peut donc aller jusqu'à l'adresse #A000, ce qui donne environ 40 ko de libres en tout.
On dispose donc en HIRES des lignes 0 à 133 et en TEXT des lignes 16 à 25. Attention, en graphismes à ne pas dépasser la ligne 133, car ensuite vous dessineriez directement dans la zone de définition des caractères du premier clavier.
Une partie de ces caractères (#B500 - #B52D) est d'ailleurs "présente à l'écran" lignes 134 et 135, et, vu la faible valeur de ces octets, ceux-ci sont interprétés comme des caractères de contrôle (Paper ou INK) qui provoquent des parasites à l'écran. La ligne n° 36 du programme se charge de les faire disparaître en mettant les bits 6 à 1 et en mettant couleur encre-couleur papier sur cette ligne.
Enfin, la dernière ligne de l'écran n'est pas entièrement condamnée car, mis à part l'octet pris par le code "31", le reste est accessible par PLOT et peut donc servir de ligne d'état comme dans le programme ci-joint.

```
0 REM------------------------------
1 REM
2 REM    Ecran  Semi-Graphique.
3 REM     Pour Oric-1 & ATMOS.
4 REM
```

CARTE MEMOIRE DANS LES 3 MODES ECRAN

TEXT

- EFFF
- ROM (BASIC) — C000
- réservé — BFE0
- Ecran Texte — BB80
- 2ᵉ clavier — B800
- 1ᵉʳ clavier — B400
- BASIC Si Grab
- Programme Basic

HIRES

- FFFF
- ROM (BASIC) — C000
- réservé — BFE0
- ECRAN HIRES (200 lignes + 3 de texte) — A000
- 2ᵉ clavier — 9C00
- 1ᵉʳ clavier — 9800
- Programme Basic

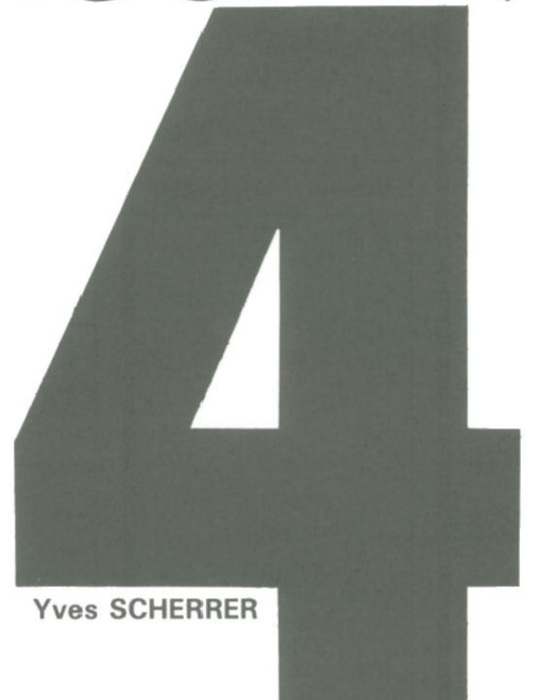
MIXTE

- FFFF
- ROM (BASIC) — C000
- réservé — BFE0
- TEXT — BE28
- zone libre — BB80
- 2ᵉ clavier — B800
- 1ᵉʳ clavier — B400
- ECRAN HIRES 133 lignes — A000
- Programme Basic

4 libre ou DOS
3 entrée/sortie
2 Ptrs + adr dvs
1 pile
0 tampon + ptrs

```
0 REM Legal Thierry      ? Aout 1985
2 REM
10 REM-------------------------------
12 REM  Programme de demonstration
14 REM-------------------------------
16 :
18 HIRES :TEXT :PAPER 4 :INK 7 :CLS
20 PLOT0,26,0:PLOT1,26,2:PLOT 2,26,96
22 PLOT 3,26,31 ' Affiche "Esc + Del"
24 PLOT 5,26,"LEGAL Thierry"
26 REM Modifier 28 et 30 si Oric-1.
28 DOKE 632,48720 :DOKE 634,48680
30 DOKE 636,360   :POKE 638,10
32 REM Efface "parasites" de l'ecran.
34 CLS  :FOR I=#B4F1 TO #B52D
36 POKE I,PEEK(I) OR 64 :NEXT '64=#40
38 POKE #200,3 'Autorise Cmds Hires
40 CURSET 0,0,0  :FILL 134,40,64
42 REM Trace du contour de l'ecran.
44 CURSET 0,134,1 :DRAW 0,-134,1
46 DRAW 239,0,1  :DRAW 0,134,1
48 POKE #B4F0,23 'Ligne 134 Blanche
50 DOKE #B518,0  'Delete Ligne 135
52 :
54 DOKE #A6,#A000 :CLEAR 'Himem #A000
56 F=(DEEK(#A0)-#400)+(#BE28-#3800)
58 PRINTHEX$(FRE(0)+F),"Octets Libres
60 :
100 CURSET 0,67,1 :DRAW 239,0,1
102 K=PI/60 :FOR X=0 TO 239 STEP 2
104 CURSET X,67-60*COS(X*K),1
106 CURSET X,67-60*SIN(X*K),1 :NEXT
108 ZAP :LIST 100-120
110 :
200 '-------------------------------
202 'Version Minimale pour Semi-Graph
204 '-------------------------------
206 ' sur Oric Atmos.
208 '
210 TEXT :PLOT 0,26,31
212 DOKE 632,48720 :DOKE 634,48680
214 DOKE 636,360   :POKE 638,10 :CLS
216 '
218 ' et sur Oric-1.
220 '
222 TEXT :PLOT 0,26,31
224 DOKE 621,48640 :POKE 623,10 :CLS
226 '
228 '
230 ' puis ...
232 '
234 '  POKE #200,3  pour utiliser les
236 '          ordres graphiques Hires
238 '
240 'et POKE #200,2 pour les ordres
242 '          en Text ou Lores comme
244 '          PLOT ou Print @ .
246 '
248 '-------------------------------
```

PUISSANCE 4



Yves SCHERRER

Aligner 4 pions... facile, penserez-vous ! Oui, mais en trois dimensions, ce n'est pas si évident que ça en a l'air.

Le jeu se joue sur un damier de 4 × 4 × 4 cases et possède la particularité de pouvoir se jouer à deux partenaires ou seul contre l'ordinateur.

Un tirage au sort détermine qui commencera le premier.

L'intérêt de PUISSANCE 4 est évident, par opposition à un banal MORPION. En effet, les pions viennent s'empiler dans les colonnes, ce qui complique la tâche pour faire une diagonale dans le sens vertical, les cases inférieures devant être pleines pour accéder à l'étage supérieur. Mais, je vous laisse juge de la difficulté en vous invitant à y jouer longuement...

Je n'entrerais pas dans le détail de la programmation, car elle est très classique pour ce type de jeu et ne possède aucune astuce particulière, le but recherché étant la difficulté du jeu et non un exercice de style en Basic. De plus, le listing se suffit de lui-même, de nombreux commentaires venant le découper.

L'honnêteté m'oblige à vous préciser que la base des calculs a été trouvée dans les revues suivantes : JEUX ET STRATEGIE et MICRO-ORDINATEURS.

Une remarque cependant concerne les DATA en fin de programme. Elles servent uniquement à faire écrire l'ordinateur comme nous le ferions avec un crayon et permettent ainsi une écriture de titre en haute résolution très originale.

Le principe de la digitalisation a été retenu pour saisir les coordonnées à l'aide d'un calque millimétré sur lequel a été fait l'original du titre. Les coordonnées ont ensuite été entrées en DATA dans l'ordre voulu. Une simple boucle REPEAT-UNTIL permettant de les afficher après indication des coordonnées de départ. Ce procédé peut être utilisé pour "digitaliser" un dessin que l'on veut reproduire sur l'écran.

Si le travail a été long et fastidieux, le résultat obtenu est à la hauteur de part l'originalité de l'écriture.

En outre, une gestion systématique des erreurs évite toute fausse manœuvre dans l'entrée des paramètres du jeu lui-même.

```
===========================================================================
===================== Liste Programme 'PUISS-4' ===================
===========================================================================


2 '----------------------------------
3 '---                            ---
4 '---   P U I S S A N C E   4    ---
5 '---                            ---
6 '---     Par  SCHERRER Yves     ---
7 '---                            ---
8 '----------------------------------
10 FOR A=40200 TO 40207:READX: POKEA,X: NEXT
20 DATA 0,7,15,30,62,60,0,0
30 GOSUB6000:'------  REGLES  ------
50 DIMA(4,4,4),Z(4,4),GA(4,4),N(20)
100 GOSUB 5000:'----  DAMIER  ------
120 FOR X=1 TO 4:FOR Y=1 TO 4:Z(X,Y)=0:GA(X,Y)=0:
125 FOR Z=1 TO 4:A(X,Y,Z)=0: NEXT Z,Y,X
130 FOR A=0 TO 20:N(A)=0: NEXT
140 N(1)=.001:N(2)=.01:N(3)=.1:N(4)=1:N(5)=10:N(10)=100:N(15)=1000:N(20)=10000
150 NC=0
160 IF NJ=2 THEN 4000
170 IF OJ=2 THEN XM=INT(RND(1)*4+1):YM=INT(RND(1)*4+1): GOTO 700
300 ' ----------------------------
310 ' --- COUP JOUEUR          ---
320 ' ----------------------------
325 NC=NC+1:IF NC=65 THEN 2500
330 SOUND1,100,9:PLAY1,0,0,0:WAIT20:PLAY0,0,0,0:POKE618,3
335 PRINT"   A vous de jouer..."
340 INPUTA$:L=LEN(A$):X=ASC(LEFT$(A$,1))-64:Y=ASC(RIGHT$(A$,1))-48
350 IFL<>2ORX<1ORX>4ORY<1ORY>4THENPRINT"Entree incorrecte !":WAIT200:GOTO330
360 IFZ(X,Y)=4 THENPRINT:PRINT"Colonne complete ! Rejouez ...":WAIT200:GOTO340
370 Z(X,Y)=Z(X,Y)+1:A(X,Y,Z(X,Y))=1
375 POKE618,2
380 GOSUB5500:'---- AFFICHAGE ------
390 IF GA(X,Y)=1 THEN Z=Z(X,Y): GOTO 2500
700 ' ----------------------------
710 ' ---     COUP ORDI        ---
720 ' ----------------------------
725 NC=NC+1:IF NC=65 THEN 2500
730 M=0:CG$="N":OG$="N":SP=2000
735 SOUND1,100,9:PLAY1,0,0,0:WAIT20:PLAY0,0,0,0
740 PRINT"   A moi de jouer !":PRINT"   Un instant, je reflechis..."
745 IF NC=2 THENXM=INT(RND(1)*4+1):YM=INT(.(1)*4+1): GOTO 790
750 FOR X=1 TO 4:FOR Y=1 TO 4:GA(X,Y)=0:T=0
760 IFZ(X,Y)=4 THEN 780
770 Z=Z(X,Y)+1:GOSUB 1000
780 NEXT Y,X
790 X=XM:Y=YM:Z(X,Y)=Z(X,Y)+1
800 A(X,Y,Z(X,Y))=5:PRINT:PRINT"Je joue en "CHR$(64+X)Y:GOSUB 5500
810 IF OG$="O" THEN Z=Z(X,Y): GOTO 2500
820 GOTO 300
1000 ' ----------------------------
1010 ' -------  S/P CALCUL  -------
1020 ' ----------------------------
1025 ' --- (1) PAR PLAN         ---
1026 ' ----------------------------
1030 S=0:FORAY=1TO4:S=S+A(X,AY,Z):NEXT:C1=0:C2=1:C3=0:GOSUB SP
1040 S=0:FORAX=1TO4:S=S+A(AX,Y,Z):NEXT:C1=1:C2=0:C3=0: GOSUB SP
1050 IF X=Y THEN S=0:FOR A=1TO4:S=S+A(A,A,Z):NEXT:C1=1:C2=1:C3=0: GOSUB SP
1060 IF X=5-Y THEN S=0:FORA=1TO4:S=S+A(A,5-A,Z):NEXT:C1=1:C2=-1:C3=0:GOSUBSP
```

```
1110 ' ---------------------------
1120 ' --- (2) DANS L'ESPACE    ---
1130 ' ---------------------------
1140 IFZ<>1ANDZ<>4THEN1250
1150 IFX=ZTHENS=0:FORAX=1TO4:S=S+A(AX,Y,AX):NEXT:C1=1:C2=0:C3=1:GOSUBSP
1160 IFX=5-ZTHENS=0:FORAX=1TO4:S=S+A(AX,Y,5-AX):NEXT:C1=1:C2=0:C3=-1:GOSUBSP
1170 IFY=ZTHENS=0:FORAY=1TO4:S=S+A(X,AY,AY):NEXT:C1=0:C2=1:C3=1:GOSUBSP
1180 IFY=5-ZTHENS=0:FORAY=1TO4:S=S+A(X,AY,5-AY):NEXT:C1=0:C2=-1:C3=1:GOSUBSP
1190 IFX=ZANDY=ZTHENS=0:FORA=1TO4:S=S+A(A,A,A):NEXT:C1=1:C2=1:C3=1:GOSUBSP
1200 IFNOT(X=5-ZANDY=5-Z)THEN1210
1205 S=0:FORA=1TO4:S=S+A(5-A,5-A,A):NEXT:C1=-1:C2=-1:C3=1:GOSUBSP
1210 IFNOT(X=5-ZANDY=Z)THEN1220
1215 S=0:FORA=1TO4:S=S+A(5-A,A,A):NEXT:C1=-1:C2=1:C3=1:GOSUBSP
1220 IFNOT(X=ZANDY=5-Z)THEN1240
1225 S=0:FORA=1TO4:S=S+A(A,5-A,A):NEXT:C1=1:C2=-1:C3=1:GOSUBSP
1240 '-----------------------------
1250 IFZ<>2ANDZ<>3THEN1350
1260 IFY=ZTHENS=0:FORAY=1TO4:S=S+A(X,AY,AY):NEXT:C1=0:C2=1:C3=1:GOSUBSP
1270 IFY=5-ZTHENS=0:FORAY=1TO4:S=S+A(X,AY,5-AY):NEXT:C1=0:C2=1:C3=-1:GOSUBSP
1280 IFX=ZTHENS=0:FORAX=1TO4:S=S+A(AX,Y,AX):NEXT:C1=1:C2=0:C3=1:GOSUBSP
1290 IFX=5-ZTHENS=0:FORAX=1TO4:S=S+A(AX,Y,5-AX):NEXT:C1=1:C2=0:C3=-1:GOSUBSP
1300 IFX=YANDY=ZTHENS=0:FORA=1TO4:S=S+A(A,A,A):NEXT:C1=1:C2=1:C3=1:GOSUBSP
1310 IFNOT(X=YANDY=5-Z)THEN1320
1315 S=0:FORA=1TO4:S=S+A(A,A,5-A):NEXT:C1=1:C2=1:C3=-1:GOSUBSP
1320 IFNOT(X=5-YANDY=Z)THEN1330
1325 S=0:FORA=1TO4:S=S+A(A,5-A,5-A):NEXT:C1=1:C2=-1:C3=-1:GOSUBSP
1330 IFNOT(X=5-YANDY=5-Z)THEN1340
1335 S=0:FORA=1TO4:S=S+A(A,5-A,A):NEXT:C1=1:C2=-1:C3=1:GOSUBSP
1340 ' -----------------------------
1350 ' --- ORTHOGONALEMENT AU PLAN -
1360 ' -----------------------------
1370 S=0:FORAZ=1TO4:S=S+A(X,Y,AZ):NEXT:C1=0:C2=0:C3=1:GOSUBSP
1400 RETURN
2000 ' -----------------------------
2010 ' --- SOUS PROGRAMME ANALYSE --
2020 ' -----------------------------
2025 IF OG$="O" THEN 2090
2030 IF N(S)=.1 THEN GA(X,Y)=1:CG$="O":YM=Y:XM=X
2040 T=T+N(S)
2050 IF T-INT(T)>=.02ANDCG$<>"O" THEN XM=X:YM=Y:CG$="BIENTOT"
2070 IFCG$="N" AND T>M THEN M=T:YM=Y:XM=X
2080 IF T>=1000 THEN XM=X:YM=Y:OG$="O"
2090 RETURN
2500 ' -----------------------------
2510 ' ---    FIN DE PARTIE       ---
2520 ' -----------------------------
2525 POKE618,2
2530 IF NC=65 THEN 3150
2540 IF OG$="O" THEN PRINT:PRINTCHR$(133)+"Desole... mais je gagne !"
2550 IF OG$="N" THEN PRINT:PRINTCHR$(133)+"Bravo, vous avez gagne !"
2555 WAIT 300
2560 SP=3000:GOSUB 1000
3000 ' -----------------------------
3010 ' - RECHERCHE LIGNE GAGNANTE --
3020 ' -----------------------------
3030 IF S<>4 AND S<>20 THEN RETURN
3032 IF NJ=1 THEN 3040
3035 PRINTCHR$(133)+"Bravo "NO$(OJ)" !...Vous avez gagne!"
3040 POP:POP
3050 REPEAT:X=X-C1:Y=Y-C2:Z=Z-C3
3060 UNTIL X=0 OR Y=0 OR Z=0
3061 X=X+C1:Y=Y+C2:Z=Z+C3
3065 O=2:FOR K=1 TO 2:O=O+1
```

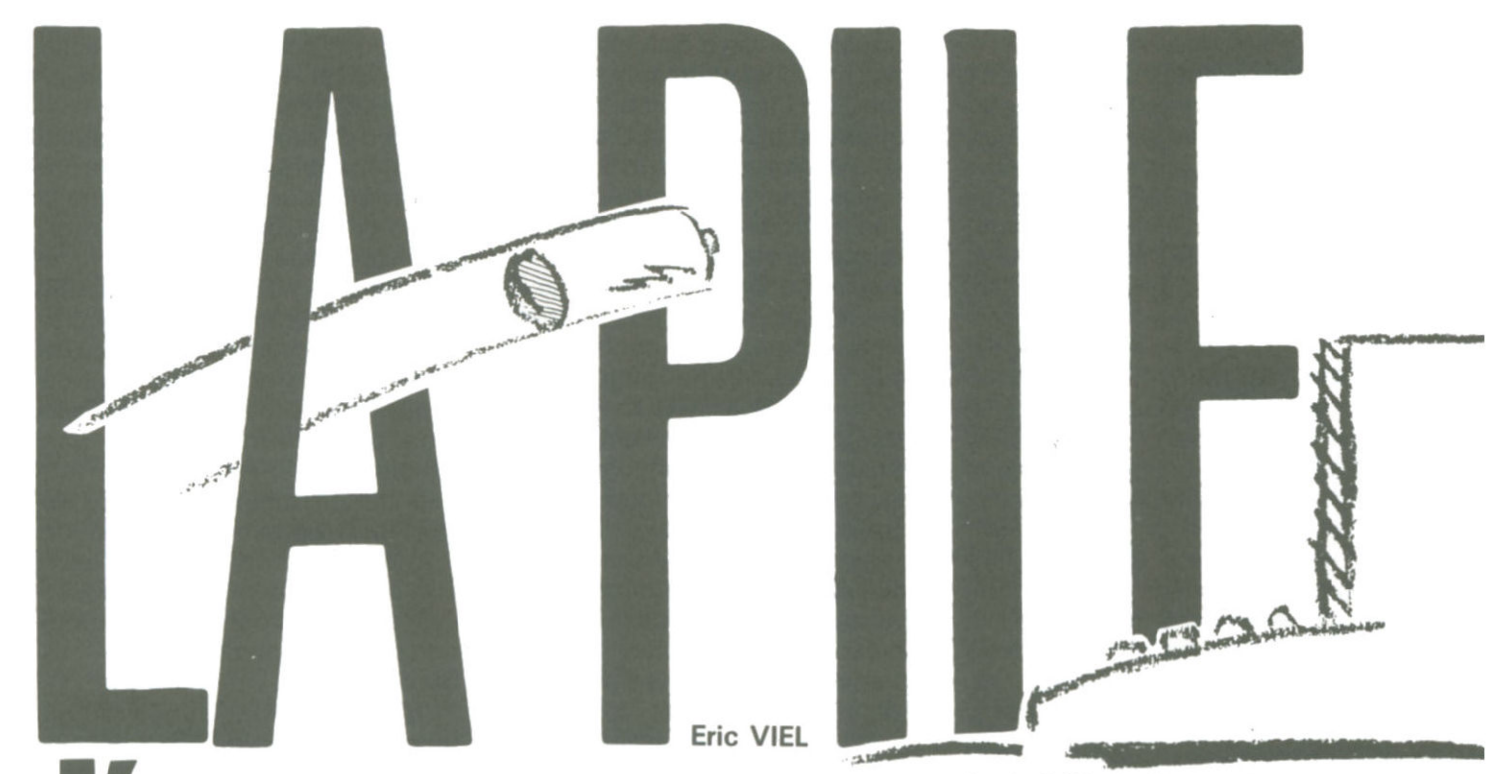
```
3070 FOR C=0 TO 7
3075 NO=1
3080 FOR N=0 TO 3:X2=X+C1*N:Y2=Y+C2*N:Z2=Z+C3*N
3090 CURSET30*X2+12*(Y2-1),236-9*Y2-47*Z2,3:FILL6,1,C
3095 NO=NO+2:MUSIC1,0,NO,9:WAIT 5
3100 NEXT N:WAIT 50
3110 NEXT C:NEXT K
3115 PLAY0,0,0,0
3120 FOR N=0 TO 3:X2=X+C1*N:Y2=Y+C2*N:Z2=Z+C3*N
3130 CURSET30*X2+12*(Y2-1),236-9*Y2-47*Z2,3:FILL6,1,1
3140 NEXT N
3150 IFNC=65THENPRINT:PRINTCHR$(130)+" Match nul !..."ELSE GOTO 3160
3151 PRINTCHR$(134)+"Une autre partie s'impose !"
3155 WAIT 200:CLS:POKE618,2:
3160 A$="**  Voulez-vous rejouer ?  **"
3170 X=0:PRINT:REPEAT
3180 PRINT RIGHT$(A$,37-X) + LEFT$(A$,X):WAIT 10
3182 PRINTCHR$(11);CHR$(129);
3185 X=X+1: IF X=38 THEN X=0
3190 B$=KEY$: UNTIL B$="O" OR B$="N"
3200 IFB$="O"THENPOKE618,3:GOSUB6140:GOTO100
3210 CLS:PRINT;CHR$(130)+CHR$(140)+"Au revoir...":WAIT500
3211 POKE618,2:TEXT:PAPER0:INK6:CLS
3215 CALL#F8D0
3220 END
4000 ' -----------------------------
4010 ' ---     A   2   JOUEURS     ---
4020 ' -----------------------------
4025 OJ=1
4030 NC=NC+1:IF NC=65 THEN 3500
4035 SOUND1,100,9:PLAY1,0,0,0:WAIT20:PLAY0,0,0,0
4040 PRINT:PRINT"   A  "NO$(OJ)"  de jouer !"
4050 INPUTA$:L=LEN(A$):X=ASC(LEFT$(A$,1))-64:Y=ASC(RIGHT$(A$,1))-48
4060 IFL<>2ORX<1ORX>4ORY<1ORY>4THENPRINT"Entree incorrecte !":WAIT200:GOTO4035
4070 IFZ(X,Y)=4THENPRINT:PRINT"Colonne complete ! Rejouez ...":WAIT200:GOTO4050
4080 Z(X,Y)=Z(X,Y)+1:Z=Z(X,Y):A(X,Y,Z)=-1*(OJ=1)-5*(OJ=2)
4090 GOSUB 5500:REM----------------------------   AFFICHAGE PION   -------
4100 SP=3000:GOSUB1000
4110 IF OJ=1 THEN OJ=2 ELSE OJ=1
4120 GOTO 4030
5000 ' -----------------------------
5010 ' ---        DAMIER          ---
5020 ' -----------------------------
5025 POKE618,2:INK3
5030 HIRES:PAPER0:INK4
5040 FOR A=0 TO 3
5050 CURSET66,10+47*A,1
5060 FOR B=0 TO 4:DRAW120,0,1:CURMOV-132,9,1:NEXT B
5070 NEXT A
5080 FOR A=0 TO 3:CURSET66,10+47*A,1:FOR B=0 TO 4:DRAW-48,36,1:CURMOV78,-36,0
5085 NEXT B,A
5090 CURSET30,190,3:FOR A=0 TO 3:CHAR 65+A,0,1:CURMOV30,0,3:NEXT A
5100 CURSET13,179,3:FOR A=0 TO 3:CHAR49+A,0,1:CURMOV12,-9,3: NEXT A
5110 FOR A=40200 TO 40207:READX: POKEA,X: NEXTA:RESTORE
5125 CURSET200,60,3:FILL8,1,5:CURSET210,60,3:CHAR33,1,1
5130 CURSET170,75,3:FILL8,1,5:CURSET175,75,3
5140 FOR N=1 TO LEN(NO$(1)):CHARASC(MID$(NO$(1),N,1)),0,1:CURMOV7,0,3:NEXT N
5145 CURSET200,110,3:FILL8,1,2:CURSET210,110,3:CHAR33,1,1
5150 CURSET170,125,3:FILL8,1,2:CURSET175,125,3
5160 FOR N=1 TO LEN(NO$(2)):CHARASC(MID$(NO$(2),N,1)),0,1:CURMOV7,0,3:NEXT N
5161 CURSET40,0,3:FILL8,1,1
5162 W$="1985 Scherrer Yves.":CURSET45,0,3:CHAR96,0,1:CURMOV14,0,3
5164 FOR N=1 TO LEN(W$):CHARASC(MID$(W$,N,1)),0,1:CURMOV7,0,3:NEXT N
```

```
5170 POKE618,3:RETURN
5400 ' ----------------------------
5410 ' ---    AFFICHAGE  PION    ---
5500 ' ----------------------------
5505 SO=400:IF NC/2=INT(NC/2)THEN C=2ELSEC=5
5510 FOR N=4 TO Z(X,Y) STEP -1
5520 X1=30*X+12*Y-6:Y1=235-9*Y-47*N
5530 CURSETX1+6,Y1+1,3:FILL6,1,4:CURSETX1-6,Y1+1,3:FILL6,1,C
5540 CURSETX1,Y1,3:CHAR33,1,1:SOUND1,SO,0:PLAY1,0,1,400:WAIT10
5550 SO=SO+50
5560 IFN<>Z(X,Y) THEN FILL8,1,64
5570 NEXT N
5580 RETURN
6000 ' ----------------------------
6001 ' ---   PRESENTATION        ---
6002 ' ----------------------------
6010 HIRES:PAPER6:INK4:POKE618,2:GOSUB30000:
6015 PLAY1,0,0,0:A=10
6020 FOR N=15 TO 50 STEP2:CURSET220,N,3:FILL2,1,22:CURSET18,N,0:FILL2,1,1
6025 SOUND 1,A,6:A=A+.5:CURMOV6,-2,3:FILL2,1,16:NEXT N:INK6
6030 FOR C=0 TO 3
6035 CURSET30,80+25*C,1
6040 FOR B=0 TO 4:DRAW85,0,1:CURMOV-88,5,0:NEXT B
6045 NEXT C
6050 FOR C=0 TO 3:CURSET30,80+25*C,1:FOR B=0 TO 4:DRAW-13,19,1:CURMOV35,-19,0
6055 NEXT B,C
6060 FOR N=70 TO190 STEP2:CURSET120,N,3:FILL2,1,22:CURSET10,N,0:FILL2,1,5
6065 SOUND 1,A,6:A=A+.5:CURMOV6,-2,3:FILL2,1,16:CURMOV120,-2,3:FILL2,1,6:NEXT
6100 A1$="PUISSANCE 4":A2$="EN TROIS"
6105 A4$="DIMENSIONS !"
6110 CURSET140,110,3:A$=A1$:GOSUB8000:CURSET155,130,3:A$=A2$:GOSUB8000
6115 CURSET140,150,3:A$=A4$:GOSUB8000
6120 FOR N=90 TO 180 STEP2:CURSET239,N,3:FILL2,1,22:CURSET130,N,0:FILL2,1,3
6125 SOUND1,A,6:A=A+.5:CURMOV6,-2,3:FILL2,1,20
6130 NEXT N:PLAY0,0,0,0:PRINT
6131 PRINTCHR$(131)+CHR$(140)+"              Joli non.?!":WAIT400
6135 ' -------- REGLES ------------
6140 TEXT:INK6:PAPER0:ZAP:POKE618,2
6141 Z$="     "+CHR$(96)+" 1984 Yves SCHERRER          "
6142 POKE48000,5:FORA=7TO39
6143 Y$=MID$(Z$,A-6,1):POKE48000+A,ASC(Y$):NEXT A
6144 X$=CHR$(4):Z$=CHR$(27):PRINT
6145 PRINTX$;Z$;"J";Z$;"C";Z$;"T---------------------------------";X$
6146 PRINT
6150 PRINTX$;Z$;"J";Z$;"C";Z$;"T---    P U I S S A N C E   4   ---";X$
6155 PRINT:POKE618,2
6160 PRINTX$;Z$;"J";Z$;"C";Z$;"T---------------------------------";X$
6165 PRINT
6170 PRINT"   Le jeu consiste a aligner"+CHR$(133)+"4"+CHR$(134)+"de ses"
6175 PRINT"pions sur un damier"+CHR$(130)+"4x4x4"+CHR$(134)+". La partie"
6180 PRINT"prend fin des que l'un des joueurs a"
6185 PRINT"reussi un alignement. "
6190 PRINT"Vous pouvez jouer contre l'ORIC"+CHR$(131)+"[1]";CHR$(134);
6191 PRINT"ou a 2 joueurs"+CHR$(131)+"[2]"
6195 PRINT"Quelle option choisissez-vous"+CHR$(129)+"(1/2)"+CHR$(134)+"?":
6196 POKE618,3
6197 GETA$:SOUND1,100,9:PLAY1,0,0,0:WAIT20:PLAY0,0,0,0
6200 NJ=VAL(A$):IFNJ<>1ANDNJ<>2THEN6197
6210 PRINT:PRINTCHR$(133)"Tirage au sort :":POKE618,2
6215 IFNJ=2THEN6250ELSE:PRINT"Choisissez entre pair et impair:      ";
6217 POKE618,3
6220 GOSUB 8200:POKE618,3
6230 GOTO6300
```

```
6250 K=PEEK(#276):IFINT(K/2)=K/2THENK$="droite":GOTO6260:ELSEK$="gauche"
6260 PRINT"Que le joueur de "K$" choisisse":PRINT"entre pair et impair.";
6270 GOSUB8200
6300 PRINT:IF NJ=2 THEN POKE618,3:GOTO6350
6310 PRINT"Votre prenom (8 lettres maximum)"
6320 GOSUB8100:NO$(OJ)=A$
6330 NO$(-1*(OJ=2)-2*(OJ=1))="   ORIC"
6340 GOTO6380
6350 PRINT"Entrez vos prenom (8 lettres maximum)";
6360 PRINT"   Joueur 1 : ";:GOSUB8100:NO$(1)=A$
6370 PRINT"   Joueur 2 : ";:GOSUB8100:NO$(2)=A$
6380 POKE618,2:WAIT100:CLS:PRINT:PRINT:PRINT:PRINT:PRINT:
6385 PRINTX$;Z$;"J";Z$;"CBonne chance ........";X$
6390 WAIT150:RETURN
8000 ' --------------------------
8001 ' - AFFICHAGE MOTS -> HIRES -
8002 ' --------------------------
8010 FOR N=1 TO LEN(A$):CHARASC(MID$(A$,N,1)),0,1:CURMOV8,0,3:NEXT
8020 RETURN
8100 ' -----------------------------
8101 ' ---- CAS DE PRENOM COMP. ----
8102 ' -----------------------------
8105 PRINTCHR$(129)
8106 INPUTA$:SOUND1,100,9:PLAY1,0,0,0:WAIT20:PLAY0,0,0,0
8110 REPEAT:IF LEFT$(A$,1)=" " THEN A$=RIGHT$(A$,LEN(A$)-1)
8120 UNTIL LEFT$(A$,1)<>" ":IF LEN(A$)>8 THENA$=LEFT$(A$,8)
8125 L=INT((8-LEN(A$))/2):IF L=0 THEN 8140
8130 FOR N=0 TO L:A$=" "+A$:NEXT N
8140 RETURN
8200 ' -----------------------------
8201 ' ---  TIRAGE AU SORT       ---
8202 ' -----------------------------
8203 POKE618,3:GETA$:SOUND1,100,9:PLAY1,0,0,0:WAIT20:PLAY0,0,0,0:POKE618,2
8205 IFA$<>"P"ANDA$<>"I"THEN8203 ELSEPRINTCHR$(129)A$
8210 N=PEEK(#276):FORK=1TO N:KK=RND(1):NEXT
8220 PLOT4,18,3:PLOT10,18,16:PLOT5,18,20
8230 FORK=1TO10:KK=INT(RND(1)*10):PLOT6,18,STR$(KK):WAIT10
8235 SOUND1,100,9:PLAY1,0,0,0:WAIT20:PLAY0,0,0,0:NEXT:WAIT30
8240 IF(A$="P"ANDKK/2=INT(KK/2))OR(A$="I"ANDKK/2<>INT(KK/2))THENOJ=1ELSEOJ=2
8245 IFOJ=1THENA$="premier":OJ=1ELSEA$="second"
8250 PRINT:PRINT"   Vous jouerez en "A$" !"
8260 RETURN
8335 ZAP:
30000 ' ---  LOGO HIRES PRES.  ------
30010 W=450:REPEAT
30015 SOUND1,W,6
30020 READ X,Y
30030 CURSETX,Y,1:W=W-.5
30040 UNTIL X=1:PLAY0,0,0,0:RESTORE
30090 ' ------------ Y ------------>
30100 DATA30,23,31,22,32,21,33,21,34,22,34,23,34,24,34,25,34,26,34,27,34,28
30110 DATA35,29,36,30,37,30,38,29,39,21,39,22,39,23,39,24,39,25,39,26,39,27
30120 DATA39,28,39,29,39,30,39,31,39,32,39,33,39,34,39,35,39,36,39,37,39,38
30130 DATA38,39,37,40,36,40,35,39,35,38,35,37,35,36,35,35,35,34,36,33,37,32
30140 DATA38,31,40,29,41,28
30150 ' ------------ V ------------>
30160 DATA42,27,42,26,42,25,43,24,44,24,45,25,45,26,45,27,45,28,45,29,46,30
30170 DATA47,30,48,29,48,28,48,28,48,27,48,26,48,25,49,26,50,27
30180 ' ------------ E ------------>
30190 DATA51,27,52,27,53,27,54,26,54,25,53,24,52,24,51,25,51,26,51,27,51,28
30200 DATA52,29,53,30,54,30,55,30,56,29,57,28,58,27
30210 ' ------------ S ------------>
30220 DATA59,26,60,25,61,24,61,25,62,26,63,27,63,28,63,29,62,30,61,30,60,30
```

```
30230 DATA59,29
30240 ' ------------ S ------------>
30250 DATA78,21,79,22,80,21,80,20,79,19,78,19,77,19,76,20,75,21,75,22,76,23
30260 DATA77,24,78,25,79,26,80,27,80,28,79,29,78,30,77,30,76,30,75,29,75,28
30270 DATA76,27,77,28
30280 ' ------------ C ------------>
30290 DATA81,30,82,29,87,26,87,25,86,24,85,24,84,24,83,25,83,26,83,27,83,28
30300 DATA84,29,85,30,86,30,87,30,88,29
30310 ' ------------ H ------------>
30320 DATA89,28,90,27,92,24,93,23,94,22,94,21,94,20,93,19,92,19,91
30330 DATA20,91,21,91,22,91,23,91,24,91,25,91,26,91,27,91,28,91,29
30340 DATA91,30,92,26,93,25,94,25,95,26,95,27,95,28,96,29,97,30
30350 DATA98,30
30360 ' ------------ E ------------>
30370 DATA99,29,100,28,102,27,103,27,104,27,105,26,105,25,104,24,103,24,102
30380 DATA24,101,25,101,26,101,27,101,28,102,29,103,30,104,30,105,30,106,29
30390 DATA107,28
30400 ' ------------ R ------------>
30410 DATA108,27,109,26,109,25,108,24,110,25,111,25,111,25,112,25,113,26,113
30420 DATA27,112,28,112,29,113,30,114,30
30430 ' ------------ R ------------>
30440 DATA115,29,116,28,117,27,118,26,118,25,117,24,119,25,120,25,121,25,122
30450 DATA26,122,27,121,28,121,29,122,30,123,30
30460 ' ------------ E ------------>
30470 DATA124,29,125,28,126,27,126,27,127,27,128,27,129,26,129,25,128,24,127
30480 DATA24,126,24,125,25,125,26,125,27,125,28,126,29,127,30,128,30,129,30
30490 ' ------------ R ------------>
30500 DATA130,29,131,28,132,27,133,26,133,25,132,24,134,25,135,25,136,26,136
30510 DATA27,135,28,135,29,136,30,137,30,138,29
30520 ' ------------ P ------------>
30530 DATA150,27,151,26,152,25,153,24,153,25,153,26,153,27,153,28,153,29,153
30540 DATA30,153,31,153,32,153,33,153,34,153,35,153,36,153,37,153,38,154,25
30550 DATA155,24,156,24,157,25,157,26,157,27,157,28,157,29,158,30,159,30
30560 ' ------------ R ------------>
30570 DATA160,29,161,28,162,27,163,26,163,25,162,24,164,25,165,25,166,26,166
30580 DATA27,165,28,165,29,166,30,167,30
30590 ' ------------ O ------------>
30600 DATA168,29,169,28,170,27,171,25,171,26,171,27,171,28,171,29,172,30,173
30610 DATA30,174,29,174,28,174,27,174,26,174,25,173,24,172,24
30620 ' ------------ P ------------>
30630 DATA175,27,176,26,177,25,178,24,178,25,178,26,178,27,178,28,178,29,178
30640 DATA30,178,31,178,32,178,33,178,34,178,35,178,36,178,37,178,38,179,25
30650 DATA180,24,181,24,182,25,182,26,182,27,182,28,182,29,183,30,184,30
30660 '------------- O ------------>
30670 DATA185,29,186,28,187,27,188,25,188,26,188,27,188,28,188,29,189,30,190
30680 DATA30,191,30,192,29,192,28,192,27,192,26,192,25,191,24,190,24,190,24
30690 ' ------------ S ------------>
30700 DATA193,29,194,28,195,27,196,26,197,25,197,24,198,25,199,26,200,27,200
30710 DATA28,200,29,199,30,198,30,197,30,196,29
30720 ' ------------ E : ---------->
30730 DATA201,28,202,27,203,27,204,27,205,27,206,27,207,26,207,25,206,24,205
30740 DATA24,204,24,203,25,203,26,203,27,203,28,204,29,205,30,206,30,207,29
30750 DATA208,28,212,26,212,29,1,1,1
30760 ' ---------------------------
30780 RETURN
30800 ' ---------------------------
```

```
===========================>   13656 Octets occupes ----------------
```



LA PILE

Eric VIEL

L'ORIC, comme tout ordinateur, possède une pile, dépendante du microprocesseur 6502 qui l'équipe. Mais à quoi, cela sert-il ? A plein de choses, serait une réponse guère satisfaisante, mais ô combien justifiée !

Tout d'abord, voyons ce que c'est. La pile est un emplacement réservé de 256 octets dans la mémoire vive, en page 1 (adresses de # 100 à # 1FF). Lorsqu'en Basic, vous établissez une boucle FOR... TO... NEXT, ou que vous appelez une sous-routine (GOSUB), l'adresse d'appel est sauvegardée dans la pile, ainsi que quelques autres informations, dépendant de l'instruction demandée. Le fait d'écrire quelque chose dans la pile s'appelle empiler. De même, dépiler signifie ôter un élément de la pile. Lorsque l'on dépile, c'est le dernier élément empilé que l'on obtient : la pile est du type LIFO (acronyme de Last In, First Out = Dernier rentré, premier sorti). La figure 1 illustre le fonctionnement d'une pile LIFO. Remarquez sur le côté droit l'index noté S. C'est le pointeur de pile du 6502. (La pile ne dépend que du microprocesseur.) Il indique en permanence l'endroit où sera empilé ou dépilé le prochain élément. S est un registre au même titre que X et Y. Mais, n'allez tout de même pas vous en servir comme index d'une boucle, cela risquerait de tout planter ! D'ailleurs, les instructions qui le concernent sont nettement différentes : on ne peut donner de valeur à S. Il faut transiter par X et les instructions de transfert TXS etTSX (qui transfèrent respectivement X dans S et S dans X). Les autres instructions se rapportant à la pile sont : PHA (empile le contenu de A), PLA (transfère le sommet de la pile dans A), PHP (empile P), PLP (dépile dans P). D'autres instructions modifient également la pile, mais de façon indirecte : JSR (empile le registre-double PC), RTS qui dépile ce même registre, RTI, qui dépile PC et P (empilés par une interruption, mais là, c'est l'article du mois prochain...).

A noter également : par rapport à la figure 1, la pile est renversée. Le premier élément empilé prend place en # 1FF, le second en # 1FE, le suivant en # 1FD et ainsi de suite...

Appliquons maintenant ces formules à des petits programmes, sans grande prétention...

Voyons d'abord comment simuler le POP du Basic. Cette fonction permet d'oublier la dernière instruction GOSUB. De même, nous allons voir comment faire oublier la dernière adresse empilée par un JSR. Il suffit de ces quatre instructions :

```
TSX
INX
INX
TXS
```

Et voilà ! Cela revient à remonter le pointeur de pile de deux octets (n'oubliez pas que la pile est renversée : sur la figure 1, cela se traduirait par une descente de deux cases du pointeur S).

Continuons en visualisant l'état des différents drapeaux, contenus dans le registre P. Ce sont les listings 1 (programme désassemblé) et 1 bis (traduit en un chargeur Basic).

Nous allons transférer P sur la pile (ligne 17), le reprendre dans l'accumulateur, en faire un double afin de ne pas le modifier (ligne 19). Puis, par une série de rotations, nous allons faire tomber ses bits un à un dans la retenue C. Par un simple test BCC, nous allons afficher le nom du drapeau s'il est à 1 (ligne 32) ou un tiret, s'il est à 0 (ligne 35). En fin de routine, nous restaurons P à sa valeur initiale (ligne 55).

Voyons maintenant quelque

chose d'un peu plus corsé : comment faire des JSR relogeables (indépendants de l'adresse d'implantation de la routine) ? En effet, si on peut aisément remplacer les JMP par CLC puis par un BCC, il n'en va pas de même pour les JSR qui ont besoin d'une adresse absolue. Regardez le listing : nous plaçons à une adresse fixe, hors de la routine, le code du RTS. Au moment d'exécuter la sous-routine, nous faisons le JSR sur le code du RTS. Nous retournerons donc au programme source immédiatement. Mais, cette opération a empilé puis dépilé l'adresse d'appel. Or, comme le dépilage du PC consiste simplement à incrémenter S de 2, l'adresse n'est pas perdue. A l'aide du CLC placé avant le JSR et du BCC placé après, nous nous branchons sur la sous-routine. (L'intermédiaire PONT n'a pas besoin d'exister, mais il montre comment procéder si le déplacement est supérieur à 128. Le label PONT reprend l'instruction BCC et envoie sur la sous-routine. Afin de ne pas perturber la région où est placée cette instruction, nous plaçons, avant le label PONT, l'instruction SEC qui donnera un résultat "faux" avec BCC, donc pas de branchement.)
La sous-routine doit maintenant exploiter l'adresse dans la pile. Je vous montre deux façons possibles :
Version 1 : On transfère S dans X, on le décrémente deux fois, on retransfère X dans S. Mais le RTS ira se brancher juste après le JSR retour, sur l'instruction BCC (d'où l'intérêt de placer le CLC avant le JSR). On place donc un SEC à la fin de la sous-routine, ce qui passera l'instruction BCC sans faire le branchement.
Version 2 : On incrémente de 2 l'adresse de retour (on va directent la modifier dans la pile). Deux représente la longueur de l'instruction BCC XXXX. Puis, on décrémente de 2 le pointeur S, afin de "remémorer" l'adresse ainsi modifiée. Ainsi, lors du RTS, le retour se fera après le BCC, plus besoin du SEC.
Voilà donc deux méthodes pour rendre relogeables vos utilitaires, de façon fort simple. Exploitez-les et envoyez vos créations à la rédaction...

NOTE : Peut-être vous êtes-vous étonné de l'instruction LDA$ 0FF,X. Pourquoi ne pas utiliser le mode page zéro, ce qui économise un octet ? A cause d'un bug du microprocesseur ! Aussi surprenant que cela puisse paraître, le 6502 est buggé. Il est en effet fâché avec les retenues. Dans le cas d'une instruction LDA$ FF,X, si X est à 1, on devrait obtenir l'adresse en # 100. Hélas, non, la retenue est perdue, et vous obtenez le contenu de # 00.
De même pour les JP indirects : JMP ($5FFF), par exemple, n'ira pas se brancher à l'adresse contenue en $5FFF (poids faible) et $6000 (poids fort), mais celle contenue en $5FFF (poids faible) et $5F00 (poids fort).
Ce type de problème est d'autant plus gênant que, sur le papier, le programe tourne parfaitement.
A noter également l'utilisation peu scrupuleuse de la pile faite par certains éditeurs... Ils créent un petit programme dans la pile, permettant de charger la suite de leur soft, et stockent en haut de la pile l'adresse de lancement de leur programme. Ainsi, pas besoin d'auto-RUN. En effet, quand vous tapez CLOAD" ", le système empile l'adresse de retour dans l'interprêteur. Aussi, après chargement, cette adresse est remplacée par celle d'exécution de la routine de l'éditeur. Pas bête, non ?
Je n'irai pas jusqu'à vous fournir un programme de copie, mais je vous conseillerais simplement de réécrire une routine de chargement, inspirée de celle de la ROM, qui modifie et/ou impose son adresse de chargement, ailleurs que dans la pile. Vous n'avez alors qu'à enlever les diverses protections présentes, puis de retransférer le tout dans la pile, en modifiant le pointeur S vers une zone inoccupée de la pile, afin de ne pas modifier les adresses en haut de pile.
Au fait, la loi du 11 mars 1957, revue et corrigée le 3 juillet 1985, n'autorise que les copies de sauvegarde. A bon entendeur... Salut !

Figure 1

	a)	b)	c)	d)	e)	f)
Pile	A ←S	B ←S / A	C ←S / B / A	B ←S / A	A ←S	←S
L'accumulateur contient :		B	C	X	C	B
Instruction		PHA	PHA	PLA	PLA	PLA
L'accumulateur contient alors :		B	C	C	B	A

En d, x est une valeur quelconque, servant à bien montrer que c passe du sommet de la pile à l'accumulateur. Les opérations b), c), d), e) et f) sont effectuées avec la pile du schéma précédent.



THEORIC est une revue sérieuse, par laquelle on essaie d'apprendre au lecteur un maximum de choses sur sa machine favorite. Comme il est bon de se détendre de temps en temps, nous vous proposons de taper le programme suivant : résultat, en couverture du prochain numéro.

```
5 PAPERO:INKO
10 DOKE#306,10000:HIRES
20 READX,Y,I
25 IFI=0THEN100
30 IFI<0THENA=0:Z=1
40 IFI>0THENA=1:Z=0
50 FORK=1TOABS(I)
60 CURSETX,Y,1
70 X=X+A:Y=Y+Z
80 NEXT
90 GOTO20
100 CURSET30,70,1:DRAW50,0,1
110 CURSET100,70,1:DRAW50,0,1
115 DRAW0,95,1:DRAW-17,0,1
120 CURSET117,165,1:DRAW-51,0,1
125 CURSET47,165,1:DRAW-18,0,1
130 DRAW0,-95,1
150 END
200 DATA76,41,30,76,42,30
202 DATA75,43,02,81,43,25
204 DATA75,44,02,86,44,16
206 DATA104,44,3,74,45,04
208 DATA89,45,06,99,45,02
210 DATA 104,45,5,74,46,03
212 DATA 90,46,05,104,46,5
214 DATA 72,47,05,90,47,03
216 DATA 104,47,6,72,48,05
218 DATA 91,48,2,105,48,05
220 DATA 71,49,05,91,49,01
222 DATA 106,49,4,81,54,06
224 DATA 81,55,-2,97,52,04
226 DATA 100,53,2,101,54,2
228 DATA 102,55,-2,84,57,2
230 DATA 84,58,02,96,57,02
232 DATA 96,58,02,70,49,-3
234 DATA 71,48,-7,72,48,-6
236 DATA 73,48,-5,74,48,-6
238 DATA 75,48,-5,71,53,01
240 DATA 68,55,2,106,50,-2
242 DATA 107,50,-5,72,6,-6
244 DATA 108,50,-14,73,5,1
246 DATA 75,8,-2,108,50,-5
248 DATA 74,6,-2,109,50,-7
250 DATA 76,09,-2,77,11,-2
252 DATA 71,12,-6,72,18,-3
254 DATA 73,21,-5,74,26,-5
256 DATA 75,31,-5,76,36,-5
258 DATA47,162,-2,48,164,-2
260 DATA 49,166,1,50,167,3
262 DATA 53,168,2,55,169,7
264 DATA 61,168,2,63,167,3
266 DATA65,163,-4,64,159,-5
268 DATA 60,156,1,58,155,3
270 DATA58,156,1,57,157,-5
272 DATA54,156,3,53,157,-6
274 DATA 52,163,1,51,162,1
276 DATA 48,161,3,62,158,2
278 DATA61,156,-6,122,170,6
280 DATA128,169,2,130,167,-2
282 DATA131,166,2,135,163,-2
284 DATA133,162,2,131,163,2
286 DATA129,164,2,129,157,-7
288 DATA126,156,3,124,156,1
290 DATA122,155,3,121,156,2
292 DATA121,157,-5,119,158,2
294 DATA118,159,-6,117,163,-5
296 DATA118,168,2,120,169,2
298 DATA125,157,-5,70,165,-16
300 DATA 71,181,-5,72,186,2
302 DATA 73,187,-3,74,187,3
304 DATA 77,186,6,83,187,04
306 DATA 86,186,2,84,188,02
308 DATA85,189,-3,88,175,-11
310 DATA89,165,-10,71,190,2
312 DATA 66,191,5,62,192,04
314 DATA 60,193,2,59,194,-2
316 DATA 59,196,3,61,197,07
318 DATA 68,196,3,71,197,13
320 DATA 84,196,1,85,195,01
322 DATA 86,192,-3,92,165,-10
324 DATA 93,175,-11,94,186,2
326 DATA 95,187,-2,96,187,-5
328 DATA 95,192,-3,96,195,1
330 DATA 97,196,1,98,197,13
332 DATA 111,196,3,114,197,7
334 DATA 120,196,3,122,193,-3
336 DATA 120,193,2,111,165,-16
338 DATA 110,181,-5,108,186,2
340 DATA 105,187,4,99,186,6
342 DATA 96,187,3,107,188,2
344 DATA 108,189,1,109,190,2
346 DATA 111,191,5,116,192,4
350 DATA 85,61,01,84,62,01
352 DATA 83,63,01,82,64,01
354 DATA 81,65,-8,82,72,-2
356 DATA 83,74,01,84,75,01
358 DATA 85,76,02,86,77,03
360 DATA 89,78,04,93,77,03
362 DATA 95,76,02,97,75,01
364 DATA 98,74,01,99,73,-2
366 DATA 100,65,-8,99,64,1
368 DATA 98,63,01,97,62,01
```

```
370 DATA 96,61,01,66,65,02
372 DATA 64,66,03,62,67,03
374 DATA 61,68,02,61,69,01
376 DATA 114,65,2,115,66,3
378 DATA 117,67,3,119,68,2
380 DATA 120,69,1,65,57,-5
382 DATA 66,56,-3,67,56,-2
384 DATA 68,55,-2,69,55,01
386 DATA 66,60,-3,67,61,-2
388 DATA 68,62,-3,69,64,04
390 DATA 68,59,04,70,60,02
392 DATA 69,61,1,73,56,-10
394 DATA 110,59,4,110,60,2
396 DATA 112,61,1,109,64,5
398 DATA 113,62,2,114,61,2
400 DATA115,60,-3,116,57,-5
402 DATA115,56,-3,114,56,-2
404 DATA113,55,-2,112,55,01
406 DATA113,63,01,106,36,-5
408 DATA107,31,-5,108,26,-5
410 DATA109,20,-6,108,19,01
412 DATA110,20,01,111,19,03
414 DATA114,18,03,117,17,02
416 DATA119,16,02,121,15,01
418 DATA121,14,02,119,13,02
420 DATA116,12,03,112,11,04
422 DATA109,12,03,107,13,02
424 DATA105,14,02,104,15,01
426 DATA103,16,01,102,17,-2
428 DATA101,18,-2,100,20,-2
430 DATA99,22,-03,98,25,-02
432 DATA 74,06,-2,75,08,-2
434 DATA 76,09,-2,77,11,-2
436 DATA 78,13.-2,79,15,-2
438 DATA 80,16,-2,81,18,-2
440 DATA 82,20,-2,83,22,-4
442 DATA 84,26,02,85,25,03
444 DATA 88,24,04,92,25,03
446 DATA 94,26,02,96,27,02
448 DATA 74,65,02,76,66,03
450 DATA 78,67,02,79,68,02
452 DATA 74,65,02,76,66,03
454 DATA 106,65,1,106,64,2
456 DATA 101,68,2,102,67,2
458 DATA 103,66,4,70,51,06
460 DATA 70,52,06,70,53,07
462 DATA 70,54,06,71,55,02
464 DATA 71,56,01,74,56,01
466 DATA 83,34,-6,84,32,01
468 DATA 85,31,01,86,32,01
470 DATA 87,34,-6,84,34,03
472 DATA 84,37,03,90,34,-6
474 DATA 91,35,01,92,36,-2
476 DATA 93,38,01,94,34,-6
478 DATA 97,34,-6,98,34,04
480 DATA 98,37,02,98,39,04
482 DATA 70,51,-4,75,55,01
484 DATA 69,53,-3,111,53,-3
486 DATA 110,51,-7,109,51,-7
9999 DATA 0,0,0 ■
```

HARDCOPY SUR FASTEXT80

Bruno BOUCHAUD

Voici pour les heureux possesseurs d'une imprimante Smith Corona Fastext 80 un programme qui leur permettra de copier leurs œuvres sur papier.

Pour faire une copie d'écran, vous avez deux possibilités :
— faire un programme Basic, et il faudra attendre souvent plus d'une demi heure,
— faire un programme en Assembleur.

C'est bien entendu la seconde solution que nous avons choisie. Vous aurez ainsi votre dessin recopié en deux secondes environ.

Il faut savoir que la FASTEXT 80 se met en mode graphique en tapant LPRINTCHR$(27);''*''; CHR$(0);CHR$(X);CHR$(0) ce qui permet d'inscrire X caractères graphiques sur une ligne. Les caractères affichés représentent les 8 bits du code ASCII du caractère envoyé à l'imprimante. Ces 8 bits sont représentés verticalement, bit 0 en bas.

BITS		
7	128	
6	64	*64
5	32	
4	16	
3	8	
2	4	
1	2	
0	1	* 1
résultat		64

65 est le code ASCII du caractère A. Il suffira donc de taper LPRINT''A'' en mode graphique pour voir apparaître deux points aux emplacements présentés cidessus.

Le programme scrute donc l'écran par groupes de 8 lignes de la gauche vers la droite de l'écran.

Plusieurs routines sont utilisées :
F5C1 utilisée avec le registre A qui équivaut à LPRINTCHR$(A) ; le registe A doit contenir le code ASCII du caractère à envoyer.
F1C8 équivaut à l'instruction POINT du Basic. Elle s'utilise comme suit :
PRINT POINT(9,19) équivaut à DOKE # 2E1,9:DOKE # 2E3,19: CALL # F1C8

Le résultat est transféré en # 2E1. O signifie que le pixel est éteint et − 1 qu'il est allumé.

Tapez le programme Basic puis faites RUN. Le programme vous signalera s'il a trouvé des erreurs dans les DATA et vous donnera la ligne concernée.

Une fois le programme rentré correctement, sauvegardez le programme en Assembleur par CSAVE''HARDCOPY'',A # 7000, E # 711E

Lancez-le par CALL # 7000.

```
10 REM  ?????????????????
20 REM  ?  HARDCOPY SUR  ?
30 REM  ?  SMITH CORONA  ?
40 REM  ?   FASTEXT 80   ?
50 REM  ?                ?
60 REM  ?   ECRIT PAR    ?
70 REM  ? BOUCHAUD BRUNO ?
80 REM  ?????????????????
100 FORI=#7000TO#710ESTEP9
101 SOMME=0
103 FORJ=0TO8
110 READA$:A=VAL("#"+A$)
120 SOMME=SOMME+A
130 POKE(I+J),A
140 NEXTJ
150 READVE
160 IFVE<>SOMMETHENPING:PRINT"  ERREUR EN LIGNE : ";
    (200+(I-#7000)/9*10):STOP
170 NEXTI
175 FORI=#7117TO#711D:READA$:A=VAL("#"+A$):POKEI,A:NEXT
180 STOP
200 DATA A9,1B,20,C1,F5,A9,41,20,C1,1125
210 DATA F5,A9,08,20,C1,F5,A9,1B,20,1120
220 DATA C1,F5,A9,2A,20,C1,F5,A9,00,1288
230 DATA 20,C1,F5,A9,F0,20,C1,F5,A9,1518
240 DATA 00,20,C1,F5,A2,00,A0,00,A9,961
250 DATA 00,85,65,20,FC,70,C9,00,10,847
260 DATA 07,F0,05,18,A9,80,85,65,20,839
270 DATA FC,70,C9,00,10,09,F0,07,18,861
280 DATA A9,40,65,65,85,65,20,FC,70,1065
290 DATA C9,00,10,09,F0,07,18,A9,20,698
300 DATA 65,65,85,65,20,FC,70,C9,00,1033
310 DATA 10,09,F0,07,18,A9,10,65,65,683
320 DATA 85,65,20,FC,70,C9,00,10,09,856
330 DATA F0,07,18,A9,08,65,65,85,65,884
340 DATA 20,FC,70,C9,00,10,09,F0,07,869
350 DATA 18,A9,04,65,65,85,65,20,FC,917
360 DATA 70,C9,00,10,09,F0,07,18,A9,778
370 DATA 02,65,65,85,65,20,FC,70,C9,1035
380 DATA 00,10,09,F0,07,18,A9,01,65,567
390 DATA 65,85,65,48,8A,48,98,48,18,865
400 DATA A5,65,20,C1,F5,68,A8,68,AA,1282
410 DATA 68,E8,E0,F0,F0,07,98,E9,07,1439
420 DATA A8,4C,2C,70,18,C0,C8,F0,4E,1134
430 DATA 48,8A,48,98,48,A9,0D,20,C1,913
440 DATA F5,A9,1B,20,C1,F5,A9,2A,20,1154
450 DATA C1,F5,A9,00,20,C1,F5,A9,F0,1486
460 DATA 20,C1,F5,A9,00,20,C1,F5,68,1213
470 DATA A8,68,AA,68,A2,00,4C,2C,70,940
480 DATA 18,8E,E1,02,8C,E3,02,A9,00,931
490 DATA 8D,E2,02,8D,E4,02,48,8A,48,1022
500 DATA 98,48,20,C8,F1,68,A8,68,AA,1243
510 DATA 68,AD,E1,02,C8,60,60 ■
```

```
100 ;"HARDCOPY SUR FASTEXT 80 de #7000 a #711D"
110 ;
120 ;"Desassemble par 'SUPER DESASSEMBLEUR' C. (84) j.p. LAURENT"
130 ;
140 .0 OFSE 0000
150 .0 R017 F1C8
160 .0 R016 02E4
170 .0 R015 02E2
180 .0 R014 02E3
190 .0 R013 02E1
```

```
200 .0 R000 F5C1
210 ;
220 LDA #1B        -
230 JSR R000       :
240 LDA 'A         :
250 JSR R000       :
260 LDA #08        :
270 JSR R000       :
280 LDA #1B        : equivaut en BASIC a LPRINTCHR$(27);
290 JSR R000       :
300 LDA #2A        : "A";CHR$(8);CHR$(27);"*";CHR$(0);CH
310 JSR R000       :
320 LDA #00        : R$(240);CHR$(0)
330 JSR R000       :
340 LDA #F0        :
350 JSR R000       :
360 LDA #00        :
370 JSR R000       -
380 LDX #00        :
390 LDY #00        : mise a zero des registres et appel
400 R011           :
410 LDA #00        : du sous-programme R001.
420 STA 65         :
430 JSR R001       -
440 CMP #00        :
450 BPL R002       : Si pixel allume met 128 dans l'a-
460 BEQ R002       :
470 CLC            : dresse 65 (bit 7).
480 LDA #80        :
490 STA 65         :
500 R002           -
510 JSR R001       :
520 CMP #00        :
530 BPL R003       : Si pixel allume ajoute 64 a l'ad-
540 BEQ R003       :
550 CLC            : resse 65 (bit 6).
560 LDA '@         :
570 ADC 65         :
580 STA 65         :
590 R003           -
600 JSR R001       :
610 CMP #00        :
620 BPL R004       : Si pixel allume ajoute 32 a l'ad-
630 BEQ R004       :
640 CLC            : resse 65 (bit 5).
650 LDA #20        :
660 ADC 65         :
670 STA 65         :
680 R004           -
690 JSR R001       :
700 CMP #00        :
710 BPL R005       : Si pixel allume ajoute 16 a l'ad-
720 BEQ R005       :
730 CLC            : resse 65 (bit 4).
740 LDA #10        :
750 ADC 65         :
760 STA 65         :
770 R005           -
780 JSR R001       :
790 CMP #00        :
800 BPL R006       : Si pixel allume ajoute 8 a l'adres-
810 BEQ R006       :
820 CLC            : se 65 (bit 3).
830 LDA #08        :
840 ADC 65         :
850 STA 65         :
860 R006           -
870 JSR R001       :
880 CMP #00        :
890 BPL R007       : Si pixel allume ajoute 4 a l'adres-
900 BEQ R007       :
910 CLC            : se 65 (bit 2).
920 LDA #04        :
930 ADC 65         :
940 STA 65         :
950 R007           -
960 JSR R001       :
970 CMP #00        :
980 BPL R008       : Si pixel allume ajoute 2 a l'adres-
990 BEQ R008       :
1000 CLC           : se 65 (bit 1).
1010 LDA #02       :
1020 ADC 65        :
1030 STA 65        :
1040 R008          -
1050 JSR R001      :
1060 CMP #00       :
1070 BPL R009      : Si pixel allume ajoute 1 a l'adres-
1080 BEQ R009      :
1090 CLC           : se 65 (bit 0).
1100 LDA #01       :
1110 ADC 65        :
1120 STA 65        :
1130 R009          -
1140 PHA           :
1150 TXA           :
1160 PHA           :
1170 TYA           :
1180 PHA           :
1190 CLC           : Imprime sur l'imprimante le carac-
1200 LDA 65        :
1210 JSR R000      : tere dont le code ASCII est en 65.
1220 PLA           :
1230 TAY           :
1240 PLA           :
1250 TAX           :
1260 PLA           :
1270 INX           -
1280 CPX #F0       :
1290 BEQ R010      : Si X different de 240 alors remet
1300 TYA           :
1310 SBC #07       : Y a son etat initial.
1320 TAY           :
1330 JMP R011      :
1340 R010          -
1350 CLC           :
1360 CPY #C8       :
1370 BEQ R012      :
1380 PHA           :
1390 TXA           :
1400 PHA           :
1410 TYA           :
1420 PHA           :
1430 LDA #0D       :
1440 JSR R000      : Si X=240 alors X=0 et fait LPRINT
1450 LDA #1B       :
1460 JSR R000      : CHR$(13);CHR$(27);"*";CHR$(0);CHR
1470 LDA #2A       :
1480 JSR R000      : $(240);CHR$(0)
1490 LDA #00       :
1500 JSR R000      :
1510 LDA #F0       :
1520 JSR R000      :
1530 LDA #00       :
1540 JSR R000      :
1550 PLA           :
1560 TAY           :
1570 PLA           :
1580 TAX           :
1590 PLA           :
1600 LDX #00       :
1610 JMP R011      -
1620 R001          :
1630 CLC           :
1640 STX R013      :
1650 STY R014      :
1660 LDA #00       :
1670 STA R015      :
1680 STA R016      : Fonction POINT du BASIC avec X en
1690 PHA           :
1700 TXA           : 1E1 et Y en 2E3 et resultat en 2E1.
1710 PHA           :
1720 TYA           :
1730 PHA           :
1740 JSR R017      :
1750 PLA           :
1760 TAY           :
1770 PLA           :
1780 TAX           :
1790 PLA           :
1800 LDA R013      :
1810 INY           :
1820 RTS           :
1830 R012          :
1840 RTS           -
1850 .E ■
```

VALOCTET

Maurice LE GUYADER

Il est des programmes qui paraissent tellement simples qu'on ne penserait pas à les publier. Pourtant, lorsque le sujet est bien traité et la présentation soignée, de telles réalisations peuvent donner des idées.

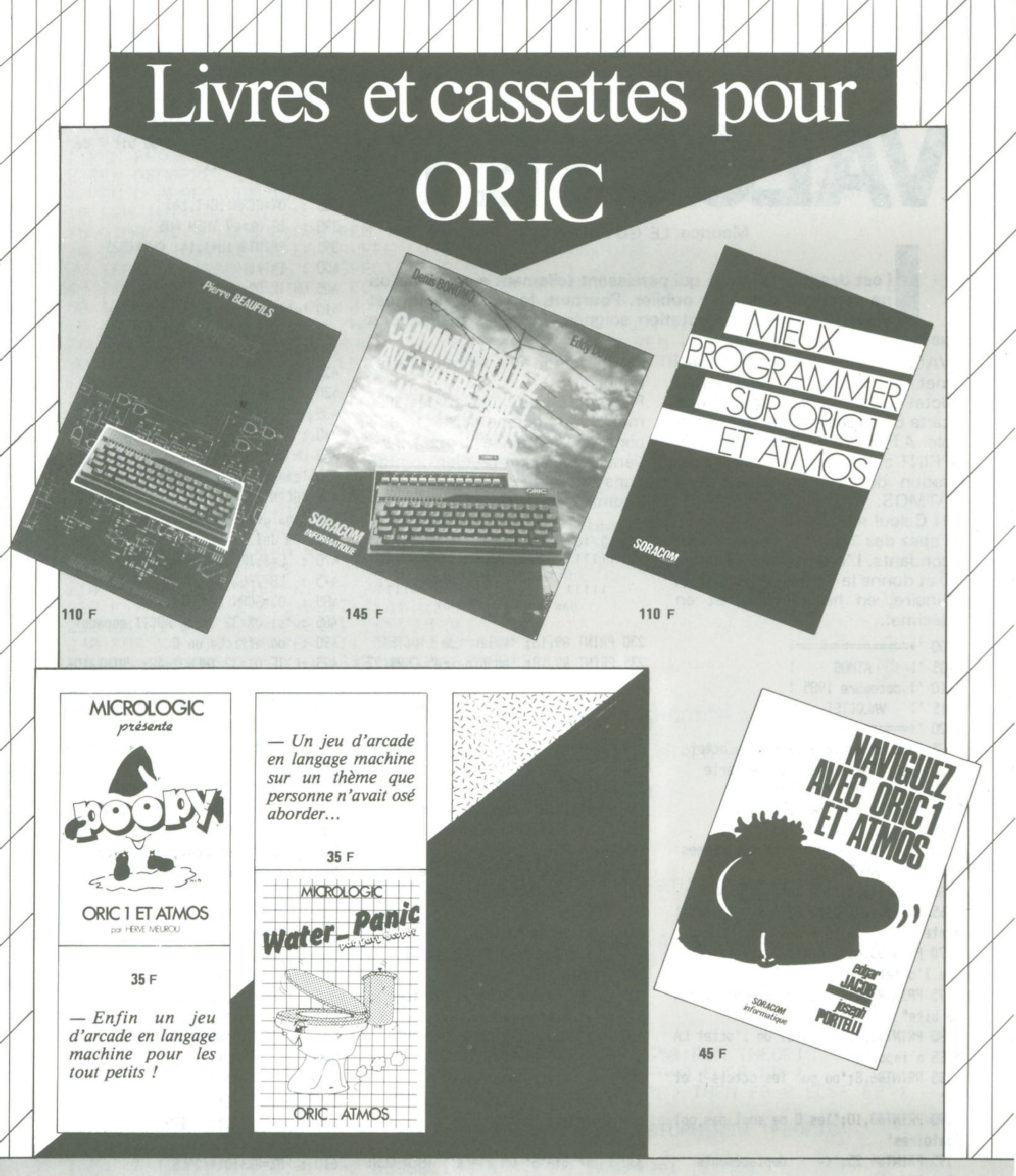
VALOCTET est de ceux-là. Il permet de calculer la valeur d'un octet pour l'adressage d'une carte 8 entrées-sorties, par exemple. ATTENTION : l'utilisation du PRINT a commercial limite l'utilisation du programme au seul ATMOS.

1) Calcul sur 1 octet :
Tapez des 1 sous les bits correspondants. L'ordinateur ajoute les 0 et donne la valeur de l'octet en binaire, en hexadécimal et en décimal.

2) Animation de lampes (effet chenillar)
Placer des 1 pour les lampes allumées, le deuxième octet étant la copie du premier, les lampes peuvent avancer ou reculer comme dans la réalité.
Exemple :

```
octet 1   octet 2            octet calculé
7654321076543210               76543210
   11111                       00011111
        (5 lampes allumées)
    11111                      10001111
        (les lampes avancent)
```

```
100 '+===============+
105 '!     ATMOS     !
110 '! decembre 1985 !
115 '!   VALOCTET    !
120 '+===============+
125 'calcul de la valeur d'un octet
130 'pour l'adressage d'une carte
135 '8 entrees 8 sorties
140 'en fonction des bits a 1
145 'et des bits a 0
150 'les REM peuvent etre supprimees
155 CLS: PAPER0: INK2
160 PRINT@14,1;"1 = lampe allumee"
165 PRINT@4,2;"0 ou ESPACE = lampe et
einte"
170 PRINT@3,4;"l'octet 2 est la copie
 de l'octet 1"
175 PRINT@6,6;"vous pouvez mettre les
 8 bits"
180 PRINT@2,7;"(1 ou 0) de l'octet LA
MPES n'importe"
185 PRINT@6,8;"ou sur les octets 1 et
 2"
190 PRINT@3,10;"les 0 ne sont pas obl
igatoires"
195 PRINT@4,25;"<=     deplacements
 0 1 SPC =>"
200 'presentation des 2 octets
205 PRINT@10,12;"-octet1--octet2-"
210 PRINT @4,13; "bits"CHR$(27)"D"CHR
$(27)"S7654321076543210 "CHR$(27)"P"
215 PRINT@8,14;CHR$(27)"G"CHR$(27)"Q"
@27,14;CHR$(27)"P"
220 'preparation de l'affichage
225 'des resultats
230 PRINT @9,16; "valeur de L'OCTET"
235 PRINT @9,18; "en binaire" CHR$(27
) "G"
240 PRINT @5,19; "en hexadecimal" CHR
$(27) "G"
245 PRINT @9,20; "en decimal" CHR$(27
) "G"
250 PRINT @8,22;"'S' pour stopper"
255 POKE#26A,3:' curseur allume
260 'initialisation des variables
265 P=10: R%=0: I=0:LS=0
270 'CLAVIER affichage des 1 et des 0
275 '          deplacement du curseur
280 REPEAT
285 : 'place le curseur
290 : 'sur le bit 7 de l'octet 1
295 :  PRINT @P,14;
300 :  GET G$:'lecture du clavier
305 : 'si fleche gauche...
310 :  IF G$=CHR$(8) AND P>10 THEN
        P=P-1: G$=""
315 : 'si fleche droite...
320 :  IF G$=CHR$(9) AND P<25 THEN
        P=P+1: G$=""
325 : 'si S...
330 :  IF G$="S" OR G$="s"  THEN CLS:
        PING: END
335 : 'si 0 ou 1 ou espace
340 :  IF G$="0" OR G$="1" OR G$=" "
THEN PRINT@ P,14; G$: IF P<25 THEN P=P
+1
345 UNTIL G$=CHR$(13):'touche RETURN
350 POKE#26A,2:'curseur eteint
355 'lit les BITS affiches en partant
360 'de la gauche et met des espaces
365 'jusqu'a la rencontre d'un 1
370 '(code ASCII 49) ou du bit 7 de
375 'l'octet droit
380 REPEAT
385 :  O%=SCRN(10+I,14)
390 :  IF O%=49 THEN 405
395 :  PRINT@ 10+I,14; CHR$(32)
400 :  I=I+1
405 UNTIL O%=49 OR I=8
410 'efface les chiffres 8 bits
415 'plus loin jusqu'au bit 0 de
420 'l'octet 2
425 P=I
430 REPEAT
435 :  PRINT@10+P+8,14;CHR$(32)
440 :  P=P+1
445 UNTIL P>7
450 'cherche par l'instruction
455 'SCRN(X,Y) le 1er bit a 1 et
460 'analyse les 7 bits suivants
465 FOR J=I TO I+7
470 :  L=J:IF L>7 THEN L=L-8
475 :  LS=7-L
480 :  O%=SCRN(10+J,14)
485 : 'si O%=32 (code ASCII espace)
490 : 'on affiche un 0
495 :  IF O%=32 THENO%=48: PRINT@10+J
,14;"0"
500 : 'affiche l'octet en binaire
505 :  PRINT@20+L,18;CHR$(O%)
510 : 'calcul du poids de l'octet
515 : 'on retranche 48 de O%
520 : 'si O% est egal a 48 (code
525 : 'ASCII de 0) alors O% vaut 0
530 : '(bit a 0)
535 : 'si O% est egal a 49 (code
540 : 'ASCII de 1) alors O% vaut 1
545 : '(bit a 1)
550 :  O%=O%-48
555 : 'additionne les valeurs
560 : 'des bits
565 : 'rappel 2↑0 = 1
570 : '       2↑1 = 2
575 : '       2↑2 = 4
580 : '       2↑3 = 8
585 : '       2↑4 = 16
590 : '       2↑5 = 32
595 : '       2↑6 = 64
600 : '       2↑7 = 128
605 : '       O%= 0ou1 (etat du bit)
610 :  R%=R%+(O%*(2^LS))
615 NEXT
620 'affiche le resultat
625 'en hexadecimal
630 PRINT@22,19;HEX$(R%)SPC(4)
635 'en decimal
640 PRINT@22,20;(R%)SPC(4)
645 PING:'alerte de fin de calcul
650 'on recommence
655 GOTO 255
```



BON DE COMMANDE

A retourner aux Editions SORACOM — La Haie de Pan — 35170 BRUZ

Nom .... **Prénom** ....

Adresse ....

Livres
☐ Communiquez avec votre ORIC-1 ATMOS .... 145 F
☐ Mieux programmer sur ORIC-1 et ATMOS .... 110 F
☐ Apprenez l'électronique sur ORIC et ATMOS .... 110 F
☐ Naviguez sur ORIC et ATMOS .... 45 F

Cassettes
☐ Poopy .... 35 F
☐ Water Panic .... 35 F

Ci-joint en règlement un chèque de .... F + 10 % de port.



(Pour les seuls numéros encore disponibles en mai 1986)

NUMERO 4

SYNTHETIS : logiciel d'accompagnement du synthétiseur vocal. **ESSAI DE LA PHS 60** interface PERITEL/UHF. Recopie graphique sur **GP 50.** Banc d'essai du modem **DIGITELEC DTL 2000. Drapeaux :** programme de jeu. **Interfaces joystick :** comparatif.

NUMERO 6

Essai de l'**XL DOS. AUTOVERIF** pour améliorer la fiabilité du chargement cassette. Essai de la carte **8 E/S MAGECO. VOICE :** ORIC parle sans interface. **RENUM-DESTRUC** renumérotation et suppression de lignes Basic. **Les interruptions :** une horloge. **Master-Mind :** jeu. **ANADIS :** analyse de disquettes au format MICRODISC. La **Jasmin-Printer** (Brother M1009) à l'essai et recopie d'écran pour cette imprimante.

NUMERO 7

BLURK'S : jeu. **Editeur de caractères** pour redéfinir le jeu de caractères. **Calculs en langage machine** pour aller plus vite. **Lecture et Ecriture de secteurs** sur Jasmin ou comment réparer une disquette. Tour d'horizon des **Moniteurs-Assembleurs** du commerce. **Le magnéto sans angoisse** ou comment améliorer le chargement des cassettes. Essai du **RANDOS.** Essai du **crayon optique.** Recopie d'écran sur **MCP 40.** L'interface **TVS.**

NUMERO 8

Une **alimentation secourue** pour se jouer des coupures de courant. L'ORIC et les **SPRITES.** Un **coffret** pour ORIC : tout en un. Comment brancher ORIC sur un **moniteur monochrome. Domptez votre MICRODISC.** Modifier le **T-DOS.** Logiciel de **Facturation.** Implantez du binaire dans des DATA : **CREADATA.**

NUMERO 10

Essai du modem **TONIC.** Le **Basic Français.** Utilitaires pour vous plaire (1). Modifications du **MONITEUR 1.0** de LORICIELS. **CROCKY :** jeu. **Compte bancaire :** gérez-le sur MICRODISC (1). Compatibilité **DTL 2000** et **JASMIN.**

NUMERO 11

Hors-série rempli d'une multitude de programmes en tous genres.

NUMERO 12

Ecrivez votre jeu d'aventure : **COLDITZ.** Le **DTL 2000 +** un serveur pas cher. **Schéma** électronique de **l'ORIC.** Essai de la table graphique **GRAPHISCOP.** Le partage **mémoire** de l'ORIC. Utilitaires pour vous plaire (2). Un **Moniteur-Assembleur.** Compte bancaire (suite et fin).

NUMERO 13

Transfert du **FORTH** sur MICRODISC. **TANDEM 13** un logiciel pour le DTL 2000. Jouez avec les **couleurs** de l'ORIC. Utilitaires pour vous plaire (3). Créer des **fenêtres** sur ORIC. **MICRODISC et JASMIN :** ils communiquent. **City on Raid :** jeu.

NUMERO 14

Le **SEDORIC** à la loupe. Un **éditeur de caractères** en langage machine. **CONJUGAISON :** ORIC à l'école. Sortir le **son sur le téléviseur. Fenêtres** (fin). Calculez votre **Epargne-Logement. Scrolling** latéral. Sauvegarde des **variables** avec **l'ORIC-1.** Un utilitaire : **Super Désassembleur.** Jouez avec les **Mots-Cachés.**

NUMERO 15

Une imprimante à l'essai : la **MT 80S. Spirographe :** rosaces et arabesques. **TRON** les motos de lumière (jeu). Essai de la carte **256 couleurs** de Microfutur. **Editeur de secteurs** sur Jasmin. **MUSICORIC** pour que l'ORIC-1 devienne musicien. **Carte série** pour liaison avec MINITEL. Educatif : **le moteur à 4 temps.** Redéfinition de caractères pour **EASYTEXT.**

NUMERO 16

Le **TELESTRAT** arrive. **Animation graphique** améliorée. **Musique :** un choral de Bach. Eléments de programmation en Basic. Un jeu : **l'Escadron suisse. Le MINITEL :** périphérique intelligent. Deux **recopies d'écran** pour **GP 100.** Un logiciel pour les petits : **POULE.** Jasmin et Microdisc : ils communiquent.

NUMERO 17

Le **TELESTRAT** se présente. Transfert de l'écran **TEXT** en **HIRES** (1). Conversion du **binaire en DATA. ORICTEL :** logiciel d'application de la carte série Minitel. **Recopie rapide** sur **imprimante** mode **texte.** Ecriture évoluée. **LEONARD :** utilitaire de dessin (1). **Fiches de paye :** comment calculer les salaires avec ORIC. **Jaquettes** de cassettes pour **GP 50. Catalogue** amélioré sur **Jasmin.**

NUMERO 18

Un **drive 3''1/2** à l'essai. **Triphonie :** l'ORIC devient orgue. Connaissez-vous les **objets fractals ?** Le Devin : chirologie ou jeu ? Un jeu : **Frog Panic.** Pour lister un programme à partir du MINITEL (et de la carte série) : **MLIST.** Accordez vos instruments de musique avec **Fréquencemètre.** Text-Hires (fin). **Léonard** (suite). Deux recopies d'écran : **GP 550** et Brother **M1009** (Jasmin Printer). Lister l'EPROM du Microdisc.

NUMERO 19

Léonard : fin de l'utilitaire création graphique. **Modifications** de l'ATMOS et du MICRODISC. Un **mini-interpréteur.** Protection du **HIMEM.** Le tri en **langage machine.** Jeu éducatif : **EUROP'ORIC.** En savoir plus sur le **SEDORIC. Compte bancaire :** gérez-le sur **JASMIN.** Utilitaires pour vous plaire (4).

Les articles et programmes rappelés ci-dessus ne forment que l'essentiel des différents numéros de THEORIC cités. Les essais des différents logiciels, les trucs et astuces, etc. sont autant d'arguments qui devraient vous inciter à compléter votre collection de THEORIC.

COLDITZ

JEU

Vivez l'Aventure!

Ce mois-ci, nous allons continuer l'analyse détaillée du traitement des verbes. Notons que le traitement des verbes constitue la partie la plus longue d'un programme d'aventure, mais aussi que cette méthode, parfaitement structurée, est celle qui assure la plus grande rapidité d'exécution.

Marcel LEJEUNE

REMPLIR

Cette fonction qui ne s'applique qu'au bidon est traitée en 4 lignes.
900 - REM REMPLIR
910 - Si l'on tente de remplir autre chose qu'un bidon vide, on a tout faux et on reçoit le message qui va bien (ligne 2820 et suivantes).
920 - De même, on ne peut remplir le bidon ailleurs que dans la fontaine.
930 - Si toutes ces conditions sont remplies, alors le bidon l'est aussi.

TUER

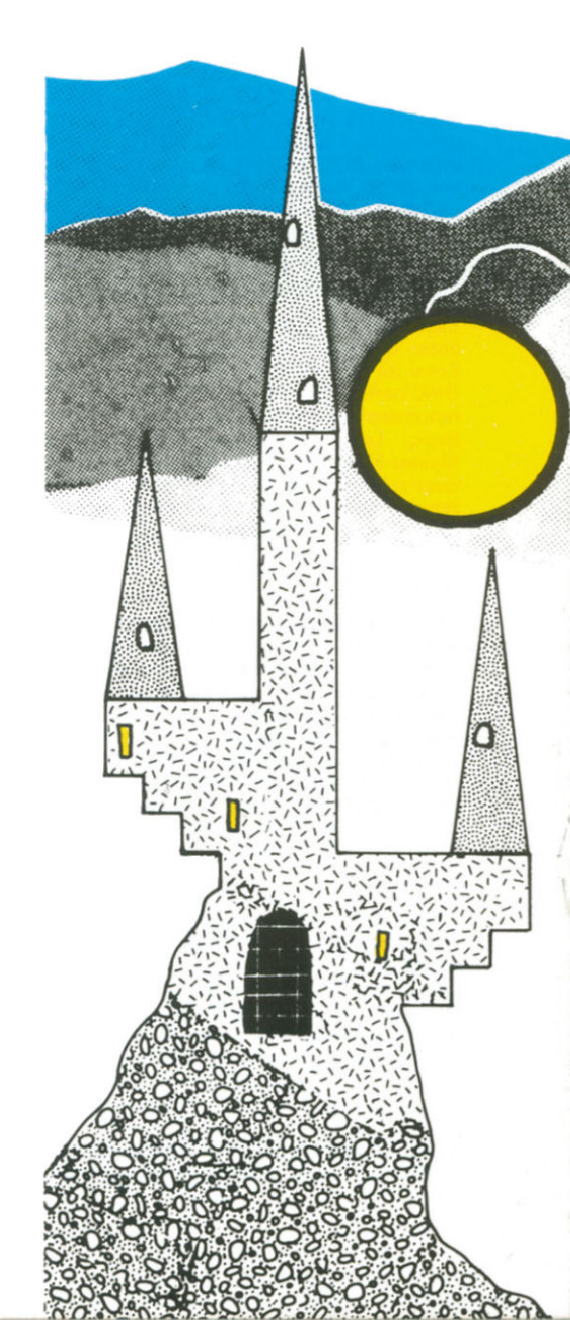
Le traitement de ce verbe est aussi simple que le précédent. On commence par la traditionnelle ligne de remarques :
950 - REM TUER
960 - On ne peut tuer que le garde qui patrouille à l'entrée de la forteresse.
980 - Il faut impérativement détenir le poignard pour pouvoir tuer le garde.

LIRE

Cette commande paraît un peu plus complexe, mais ne l'est pas en réalité ; sa longueur est due au fait que trois cas sont envisageables.
1030 - REM LIRE
1040 - Comme d'habitude, on a le droit à un message narquois si l'on essaye de lire autre chose que ce qui est écrit :
- sur le petit morceau de papier trouvé dans la salle n° 1,
- sur la feuille collée dans l'armoire du géolier,
- sur le plan du château.
1050 - Sur le petit morceau de papier, on trouve le code qui servira à ouvrir la porte du coffre situé dans le bureau du commandant.
1060 - Il faut absolument transporter le papier pour pouvoir le lire.
1080 - Idem pour le plan du château.
1065 - Le plan est sans intérêt et ne vous apprendra rien. Dommage ! Cette opération n'aura servi qu'à retarder votre progression.
1095 - On ne peut lire le message situé dans l'armoire que si l'on se trouve dans le bureau du géolier.
1100 - Ce message vous indique (en allemand, natürlich...) le numéro de l'anneau que vous devrez faire tourner pour pouvoir sortir de l'écurie où vous vous êtes laissé enfermer.

ASSOMMER

La structure de cette instruction est strictement identique à celle du verbe TUER et il serait fastidieux de vous imposer à nouveau une description détaillée. Sachez néanmoins que seule la senti-



TELESTRAT : soyez prêt à communiquer !



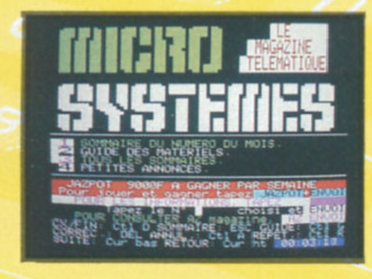
La communication a toujours été vitale à tous les niveaux de l'entreprise comme de la vie courante, et la télématique a apporté un progrès considérable au niveau de la rapidité et de la diversité de la communication. Le **MINITEL** a marqué une date dans l'approche de la télématique publique, et il était normal qu'ORIC conçoive et réalise le premier micro-ordinateur qui mette les applications télématiques à la portée de tout un chacun. Le **TELESTRAT**, outre toutes les applications que l'on peut attendre d'un micro-ordinateur moderne, vous permettra de constituer, sans aucune connaissance particulière, un serveur télématique, un répondeur téléphonique qui enregistre, affiche et écrit, une borne de communication, des liaisons à distance entre micros, bref le **TELESTRAT** est un micro-ordinateur qui **COMMUNIQUE** !

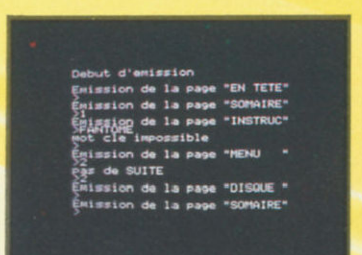
En effet, le **TELESTRAT** est équipé de nombreuses interfaces et particulièrement d'une qui permet de le raccorder à un MINITEL. Avec le **TELESTRAT**, les P&T vous offrent le moniteur et le MODEM ! Le **TELESTRAT** peut se servir du MINITEL comme d'un terminal, utiliser son écran et même son clavier, et peut ainsi communiquer par téléphone avec les serveurs, avec un autre **TELESTRAT** relié lui aussi à un MINITEL, échanger des écrans graphiques, des pages de textes, des programmes, des partitions de musique (codées MIDI bien sûr..) Le **TELESTRAT** peut également servir de terminal intelligent du MINITEL: relié à un moniteur RVB, il permet d'avoir un MINITEL en couleurs, il peut stocker sur disquette des pages VIDEOTEX ou tout ce qui peut passer par un MINITEL, et les sortir sur une imprimante tout à fait standard. Il peut aussi devenir un serveur monovoie particulièrement puissant et rapide, consultable depuis un simple MINITEL.

Des logiciels pour développer sa propre télématique

Cette utilisation télématique est bien sûr à la portée de chacun et ne demande aucune connaissance préalable d'un quelconque langage de programmation : en effet, le **TELESTRAT** est livré avec la cartouche **TELEMATIC** qui permet d'en faire un **véritable serveur monovoie privé**. A l'aide de menus simples et du curseur ou du joystick (ou de la souris en option), vous rédigez vos pages au standard VIDEOTEX (celui utilisé par le MINITEL et par de nombreux autres systèmes télématiques), incorporez des écrans graphiques, constituez des arborescences (la structure qui établit avec le correspondant un véritable dialogue à distance). Les pages sont éditées en couleurs, avec toutes les tailles de caractères et avec une grande simplicité d'utilisation : mode éditeur permettant le déplacement de blocs, l'insertion, la suppression etc.. Les arborescences sont gérées à partir de **8 commandes seulement** ! Visualisées en pleine page, elles peuvent être imprimées. Avec le MICRODISC, une arborescence peut gérer jusqu'à **1000 pages** ! Une BAL (boite à lettres) peut être insérée à n'importe quel endroit de l'arborescence avec accès libre ou accès codé par mot de passe, temporisation, possibilité de consultation à distance etc..
Le serveur ainsi constitué peut servir en mode TEST (interrogé par le minitel pour voir les conditions réelles d'utilisation, en mode SERVEUR, c'est à dire interrogé via le téléphone par un autre MINITEL ou un autre **TELESTRAT**, et enfin en BORNE DE COMMUNICATION, c'est à dire complètement autonome, avec interrogation sur place, sans passer par le téléphone ou le MINITEL.









Quelques applications ...

Dotés de telles possibilités, d'une si grande facilité d'emploi (n'importe où, il suffit de le relier à un MINITEL) , le TELESTRAT s'annonce d'ores et déjà comme un extraordinaire outil de communication, aux applications innombrables : la première et la plus inattendue est qu'il constituera un moyen de télécommunication privilégié pour tous ceux qui ont besoin d'une trace écrite (une application intéressante pour les sourds et les malentendants par exemple !). Le TELESTRAT pourra être au service de toutes les professions libérales un répondeur MINITEL, qui décroche le téléphone, informe, enregistre, imprime, peut raccrocher et même s'interroger à distance! **L'ecriture téléphonique est née !**
Les serveurs sont déja présents dans toutes les branches du commerce mais ils ont été jusqu'ici réservée aux grandes entreprises capables de posséder ou de louer un serveur. Maintenant, grace au TELESTRAT, n'importe quel commerçant pourra mettre son catalogue, son tarif, ses promotions à portée de clavier : dans la journée les clients pourront l'interroger sur place en attendant, et en dehors des heures d'ouvertures, ils pourront également s'informer via le MINITEL, laisser des messages, ou des commandes qui seront enregistrés, et lus le lendemain !
Dans un salon, un grand hotel, un grand magasin ou n'importe quel endroit où le public peut avoir à demander de l'information, le TELESTRAT utilisé en borne de communication permet de disposer 24 heures sur 24 d'un outil conversationnel, permanent, et d'une capacité de stockage et de diffusion de l'information considérable.

CARACTERISTIQUES TECHNIQUES

Processeur : 6502 , 1 MHz
Mémoire : 64 ko RAM, 48 ko ROM (MONITEUR: 8 ko, BASIC: 24 ko, MINITEL: 8Ko) . Possibilité de 64 ko suplémentaires (RAM, ROM, EPROM...)
Clavier : QWERTY / AZERTY commutable par soft.
Son : 3 canaux, + canal de bruit mixable. Sortie sur prise Péritel (son dans le moniteur)
Affichage: 8 couleurs . Mode texte : 40 colonnes, 28 lignes - Mode haute résolution : 240 x 200 points.
Alimentation : 220 V. par Boitier externe avec interrupteur. Alimente 1 TELESTRAT et 1 Drive
Dimensions : 348 x 260 x 71 mm

Interfaces : - **Vidéo** : Péritel (auto alimentée, son dans le téléviseur)
- **Standards** : Parallèle Centronics - Série RS 232 C - 2 joystick type STANDARD - Souris.
- **Mémoire de masse** : Interface cassette avec télécommande, Contrôleur de disquette (jusqu'à 4 lecteurs).
- **Original** : Péri-informatique (MINITEL) - extension MIDI (musique)
- **Divers** : Bus d'extension complet, identique à celui de l' ATMOS.

Divers : Bouton de RESET universel

LE BASIC DU TELESTRAT

BASIC d'un type nouveau : compilé à l'entrée de la ligne. La facilité d'édition d'un langage interprété, la rapidité d'un langage compilé, donc de **2 à 100 fois plus rapide** que tous les BASIC sur micro-ordinateurs. **Basic structuré** (définition de procédures, extension du vocabulaire). **Variables** : jusqu'à 16 caractères significatifs. Procédures récursives. **Editeur** : type pleine page. Correction des erreurs de syntaxe à l'entrée de la ligne. Toutes facilités de recherche, échanges d'instructions, renumérotation, destruction, liste des variables etc... Vocabulaire lié à l'édition: **LIST, SLIST, MERGE, DELETE, RENUM, NUM, SEEK, CHANGE, TRACE, CONT, DPVAR ...**
Les ordres généraux : **FOR..TO..STEP...NEXT, COUNT, ENDC, WORD, ENDW, AWORD, GOTO, GOSUB, RETURN, POP, IF...THEN...ELSE, ON...GOTO, ON...GOSUB, PRINT, SPRINT, LPRINT, USING, DATA, READ, RESTORE, DIM, GET, INPUT, POKE, DOKE, REM, RANDOM, MOVE, SWAP, ERR, ERRGOTO, RESUME, ERROR, CALL, USER, USR**
Les opérateurs et fonctions : **AND, OR, XOR, >, <, +, -, *, /, ASC, LEN, INSTR, CHR$, BIN$, HEX$, LEFT$, RIGHT$, MID$, MIDDLE$, UP$, LO$, LOB$, KEY$, SPC$, STRING$, ABS, PEEK, DEEK, INT, SGN, NOT, LN, LOG, SIN, COS, TAN, ATN, EXP, RND, SCRN, POINT, PI, FRE**
Gestion des Entrées/sorties : Tous les périphériques sont gérées de la même façon, et peuvent être échangés, modifiés etc.. 4 fenêtres d'écran, le clavier, la souris, le lecteur de disquettes, l'imprimante etc.. sont autant de périphériques. Vocabulaire lié aux E/S : **OPCH, CLCH, WIDTH, INK, PAPER, WINDOW, LWIDTH, LLINE, LFEED, LBUF, LOUT, SSPEED, SMODE, SRBUF, SEBUF, TEXT, HIRES, LORES, KEYIF, AZERTY, QWERTY, ACCENT, KEYDEF, KEYUSE, VUSER ...** La plupart des E/S sont bufférisées, c'est à dire que leur appel n'arrête pas le fonctionnement de l'ordinateur.
Le son : **SOUND, MUSIC, PLAY, SHOOT ...** *Les graphismes* : **LINE, CURSET,CURMOV, DRAW, BOX, ABOX, PAINT, CIRCLE, ELLIPSE, CHAR, SCHAR, TCURSET, TCURMOV, TDRAW, TADRAW, TABOX** etc..

LE MICRODISC

Technique : Lecteur 3 pouces double densité double face - Alimenté par l'alimentation du TELESTRAT - Capacité : 360 à 400 ko
Système d'exploitation : STRATSED, compatible avec le SEDORIC - N'empiète pas sur la mémoire utilisateur - Ultra rapide: transfert 12 à 20 Ko par seconde (écriture / lecture) - Peut gérer jusqu'à 4 lecteurs . Extrait du vocabulaire général : **SDIR,LDIR, SDIR, LOAD, SAVE, SAVEU, REN, STATUS, PROT, INIT, DEL, BACKUP, COPY, SYS, DNAME**
Gestion de fichier : Permet l'ouverture simultanée de 63 fichiers. *Séquentiel* : **OPEN, CLOSE, PUT, TAKE, APPEND, REWIND, JUMP, BUILD, TYPE, LTYPE, STYPE, EOF, FT** - *Direct* : **OPEN, CLOSE, PUT, TAKE, FIELD, LSET, RSET, SN, SL** - *Disque* (permet de travailler au niveau du secteur) : **OPEN, CLOSE, PUT, TAKE, FIELD, LSET, RSET, PMAP, SMAP, CRESEC, FRSEC**

Fabriqué et garanti UN AN pièces et main d'œuvre par



INTERNATIONAL

Siège social :
39 Rue Victor Massé
75009 - PARIS
Tel. (1) 42 81 20 02

Unité de production :
Boite Postale 30
14 140 . LIVAROT

Démonstration sur serveur MINITEL : 42 81 22 72

TELESTRAT

ATTENTION !
Si vous tournez cette page, vous vous retrouverez à jamais engagé dans l'aventure télématique !

ORIC
MICRODISC
ORIC IQ164
TELESTRAT

La télématique à la portée de tous !





TELESTRAT : c'est un micro-ordinateur

Pour répondre d'avance à la question : "Un micro-ordinateur, à quoi celà peut-il servir aujourd'hui ?", nous avons bien sûr développé le TELESTRAT autour de l'application télématique, mais c'est aussi et avant tout un micro-ordinateur moderne, puissant et qui peut faire tout ce que l'on peut attendre d'un bon micro comme applications domestiques ou professionnelles.

Un HYPER-BASIC hyper-puissant ...

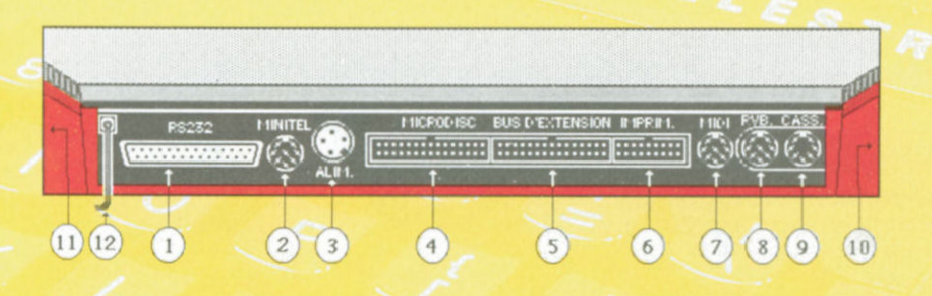
Conçu autour d'un microprocesseur 6502, le TELESTRAT dispose de l'un des plus puissants BASIC existant à l'heure actuelle sur un micro familial : **plus de 250 instructions** incluant celles du système d'exploitation du disque (STRATSED), un éditeur pleine page, 45 kOctets à la disposition de l'utilisateur, l'**HYPER-BASIC** du TELESTRAT est aussi le plus rapide : étant compilé à l'entrée de chaque ligne de programme, il s'exécute **2 à 100 fois plus vite** que n'importe quel autre programme identique non compilé.

La compatibilité Oric...

Les ORIC 1 et ATMOS sont les micro-ordinateurs familiaux qui possèdent la plus importante bibliothèque de logiciels en francais : nous avons inclus dans le TELESTRAT un programme qui le rend **100% compatible avec toutes les cassettes pour ORIC 1 ou ATMOS** et avec les disquettes non protégées ! Il dispose donc dès sa naissance de plus de 2000 programmes de jeux, programmes éducatifs, utilitaires etc.. Outre son BASIC, le TELESTRAT pourra également recevoir un Langage C, un Forth, un PASCAL, un Super Assembleur etc.. Associé au STRATSED du MICRODISC, le TELESTRAT s'annonce comme un outil de développement, d'éducation, de musique et de programmation extrêmement puissant.

Des connexions tous azimuts !

Conçu pour communiquer, le TELESTRAT est doté de nombreuses interfaces et des prises d'entrées/sorties lui permettant de se raccorder à peu près à tous les périphériques : (8) une sortie vidéo RVB-PERITEL pour un moniteur ou un téléviseur couleurs, (10&11) 2 prises pour joysticks standard ou une souris spéciale, (4) une sortie contrôleur pour brancher une ou plusieurs unités de disquettes, (9) une prise pour magnétocassette avec télécommande, (5) un BUS d'extension pour les interfaces supplémentaires, 2 ports pour cartouches ROM ou RAM (sur le dessus), (1) une entrée/sortie RS 232 (le système le plus répandu pour la communication entre micro-ordinateurs), (6) une sortie imprimante au standard CENTRONICS / parallèle (là encore le plus répandu), (7) une extension MIDI qui est le standard des instruments de musique électronique (on pourra faire jouer des synthétiseurs, stocker des programmes de synthés, des séquences musicales, écrire des partitions, enseigner le solfège etc..). Enfin, on trouve (2) la prise de l'interface MINITEL décrite plus haut et le fil de téléphone (12).



LE MICRODISC ORIC

Déja expérimenté sur la gamme ORIC, le MICRODISC est un enregistreur/lecteur de disquettes 3" (7cm) dont la rapidité et les performances sont incomparables. Doté d'un lecteur double tête, il permet une capacité de stockage de 400 kOctets. Du type lecteur esclave, il est économique et permet d'augmenter le nombre de MICRODISC jusqu'à stocker 1.6 Mégaoctets pour un prix introuvable sur le marché. Le **STRATSED,** son système d'exploitation, permet de développer des applications de fichiers, de transfert, de stockage avec une grande rapidité (12 à 20 koctets transférés par seconde en lecture comme en ecriture)





Des applications en cartouches

Les cartouches permettront de nombreuses applications en accès immédiat : les cartouches **HYPER-BASIC** et **TELEMATIC** sont déja livrées d'origine. Les cartouches **ROMORIC1** et **ROMATMOS** qui permettent la compatibilité avec les autres ordinateurs ORIC. Très prochainement, de nombreuses cartouches de langage seront disponibles : **FORTH, Compilateur C, Super-Assembleur** etc.. La cartouche **MIDI** sera livrée avec le boitier d'interface ramenant les 5 prises standard du MIDI à la prise unique du TELESTRAT. Les applications avec les synthétiseurs sont considérables : séquenceurs, mémorisation de programmes, de sons échantillonnés, écriture musicale etc.. Avec l'application télématique, on pourra même échanger de la musique par MINITEL ! Bien sur, des applications professionnelles (traitement de textes, tableurs, bases de données etc..) et des jeux seront également développés.

M.__________ Profession ____________
Adresse ___________________________
Code _____Ville __________________

Je souhaite être contacté par le point-conseil ORIC
Le plus proche de mon domicile.
J'envisage pour le TELESTRAT
- ☐ **une utilisation privée**
- ☐ **une utilisation profesionnelle**

Le TELESTRAT m'intéresse pour
- ☐ **son application en tant que micro-ordinateur**
- ☐ **une application TELEMATIQUE**

Je possède déjà un micro-ordinateur ☐ OUI ☐ NON

nelle qui détient le trousseau de clés peut être assommée et uniquement à l'aide de la barre de fer que l'on aura pris soin de ramasser dans le garage. Dès que la sentinelle est assommée, on peut lui prendre son trousseau de clés qui pemettront d'entrer dans le bureau du commandant de la forteresse.

DONNER

Cette opération, très simple également, ne s'applique qu'au bidon que l'on doit remettre au prisonnier assoiffé qui croupit au fond de sa cellule, en échange de quoi, il vous dira le code secret qui fait l'objet de votre mission.
Là encore, certaines conditions devront être remplies et c'est le rôle de la ligne 1240 : il faut être devant la cellule et transporter le bidon qui doit être plein. Dans ces conditions, le drapeau F1 passe à 1 et le drapeau F2 passe à 0 (voir leur signification dans notre précédent article).

DEMANDER

Maintenant que nous savons que le prisonnier détient le code secret et qu'il est en état de parler, il ne reste plus qu'à lui en faire la demande, qu'il s'empressera de satisfaire si vous lui êtes venu en aide en lui donnant de l'eau. Sinon, il ne pourra que marmonner des propos incompréhensibles qui ne vous seront d'aucune utilité.

COUPER

La chaîne est le seul objet qui puisse être coupé, et le test de validité de l'ordre se fait en ligne 1320.
1340 - Pour pouvoir couper la chaîne, il faut posséder une pince coupante et être là où se trouve la chaîne (cela paraît évident, mais il faut quand même effectuer un test). De plus, on s'assure que la chaîne n'a pas déjà été coupée.
1342 - Ici, vous devrez confirmer que vous voulez bien couper la chaîne avec la pince.
1345 - La trappe devient ouverte, ce qui donne un accès supplémentaire vers le nord. On modifie l'état des flags et on dit que c'est fini.

INVENTAIRE

Le rôle de cette commande est de vous donner la liste des objets que vous transportez. Nous savons déjà qu'à un instant donné, on ne peut transporter qu'un maximum de 4 objets. Pour afficher leur nom, il suffit de scruter la table de position des objets EO(x). Si l'on obtient 99, c'est que l'objet est transporté et il suffit donc d'afficher son nom et de poursuivre l'exploration de la liste.

Voilà, c'est tout pour aujourd'hui ! Dans notre prochain numéro, nous verrons les derniers verbes, et Colditz n'aura plus de secret pour vous. Il ne vous restera plus alors qu'à écrire votre propre aventure et, pourquoi pas, à la publier dans THEORIC.
A bientôt.

A SUIVRE...

BIBLIORIC

"L'ORIC A NU"
Fabrice BROCHE
(SORACOM)
151 F



Point n'est besoin de présenter l'auteur, Fabrice BROCHE : tous les passionnés d'ORIC le connaissent, car il est le père d'XL DOS et du SEDORIC (avec Denis SEBBAG). Qui d'autre que lui aurait pu réunir autant d'informations sur le système ORIC ?
L'ORIC A NU, composé directement par l'auteur en traitement de texte (merci, Easytext !) décrit de A jusqu'à Z les systèmes matériel et logiciel de l'ORIC. Schéma électronique, détail des composants, fonctionnement du clavier, des entrées-sorties cassette et imprimante, du générateur sonore ; rien n'a été oublié.
La partie "logiciel" du livre est, de loin, la plus importante. Le lecteur y découvrira la structure de l'interpréteur, le rôle des variables système et surtout le désassemblage détaillé, commenté ligne par ligne, des ROM ORIC-1 et ATMOS. Plus de problèmes pour tous ceux qui recherchent des équivalences d'adresses !
Après avoir vu ce livre, il semble difficile de se passer de la quantité d'informations contenues dans ses 420 pages.
L'ORIC A NU : le document qui faisait défaut pour bien utiliser la machine.



MIROIR SPHERIQUE

Wolfram LUTHER

Récemment, un article intitulé ''SPIEGEL-GRAPHIK'' de Martin GISSER, traitant un sujet fort attrayant, a paru dans la revue allemande ''COMPUTER-TECHNIK 1, 1986, p 104-108''.
Une sphère suspendue dans l'espace est illuminée par une source ponctuelle L, et un dessin géométrique situé dans le plan en-dessous de la sphère se réflète dans celle-ci comme dans un miroir.
Le programme, assez important, est écrit en SIMON's BASIC et tourne sur COMMODORE C64.
Comme les résultats sur écran à haute résolution sont très jolis à voir, j'ai pensé qu'il valait la peine d'adapter le programme pour notre ORIC et ai saisi l'occasion de le raccourcir et de le modifier sensiblement.

Le programme est donc devenu trois fois plus court, mais il reste un grand inconvénient, il est très long à exécuter : 7 à 8 heures. Laissez-le tourner pendant la nuit et vous serez recompensé le lendemain matin.
Expliquons d'abord la méthode employée par M. GISSER :
Le centre de la sphère K se trouve au point K=(0,0,K3), K3>=1 dans un repère X1, X2, X3 (voir figure 1), l'observateur au point Q=(Q1, Q2, Q3), Q3>0.
Placez l'écran E de projection entre l'observateur et la sphère ou même derrière la sphère comme vous le voulez.
Le vecteur C=(C1, C2, C3) relie l'observateur au centre de l'écran qui est caractérisé par les vecteurs unitaires A et B et paramétré en S, T.
Précisez maintenant les coordonnées de la source lumineuse L, le rapport largeur/hauteur de l'image HIRES sur votre écran et fournissez la largeur 2*SM de l'écran E dans le repère X1, X2, X3.
Comme l'exécution du programme demande plusieurs heures, vous pouvez introduire le pas de travail ST. Je vous conseille ST=4 pour un test rapide.
Le programme construit pour chaque pixel (X,7) de l'écran E un rayon J*Z qui se propage en ligne droite venant de l'observateur en traversant l'écran à l'endroit (X,Y) jusqu'à ce qu'il rencontre un point P du plan X3=0.
Le pixel (X,Y) aura donc la couleur du point P. Dans le cas échéant, P prendra la couleur du ciel.
Il se peut que le rayon touche au cours de son trajet, la sphère K où il subit une réflexion (voir figure 2).
Ces le cas si l'équation

(M − J*Z,   M − J*Z) = 1'

est remplie pour une valeur du paramètre J ou, autrement dit, si le déterminant
DI = MZ*MZ − ZZ*(MM − 1),
et le produit scalaire

MZ = (M,Z)

sont positifs (ou nuls) (ligne 280).
Le programme :

— calcule le point aux composantes (PX, PY, PZ) où il y a réflexion et le nouveau vecteur (Z1, Z2, Z3) (lignes 510-530) ;
— détermine le point (P1, P2, P3) d'intersection du plan (X3 = 0) et de rayon (lignes 310, 410) ;
— détecte si P est à l'ombre ou non (ligne 430) ;
— évalue la couleur du point d'intersection (lignes 610-650) ;
— et la donne au pixel (X,Y), (ligne 440).

Nous avons placé le mot THEORIC en grand dans le plan (la matrice se trouve à partir de la ligne 1000). Si l'observateur se trouve à proximité de la sphère, vous pouvez discerner l'écriture en miroir (figure 3).

Vous êtes invité à expérimenter et à introduire vos valeurs préférées de :

K, Q, C, L, TV et 2*SM

et un autre dessin en ligne 620.

Nous proposons, par exemple :

K3 = 1 ; Q: 4, −4,4; C: −4,4, −3.5; L: 40,40,10; TV = 1.45; 2*SM = 5.5 et

620 PU = ABS(P1)↑.7 + ABS(P2)↑.7:IF 2*INT(PU/2) = INT(PU) THEN FB = 0 ELSE FB = 1

ou

620 PU = P1 − INT(P1):PV = P2 − INT(P2):IF 3*PU*PV > PU*PU + PV*PV THEN FB = 0 ELSE FB = 1.

Les figures 4 et 5 montrent d'autres échantillons.

Dans un deuxième programme, nous avons introduit une deuxième sphère de rayon 2 en-dessous de la première et supprimé le mot THEORIC (voir figure 6 et programme : Deux sphères).

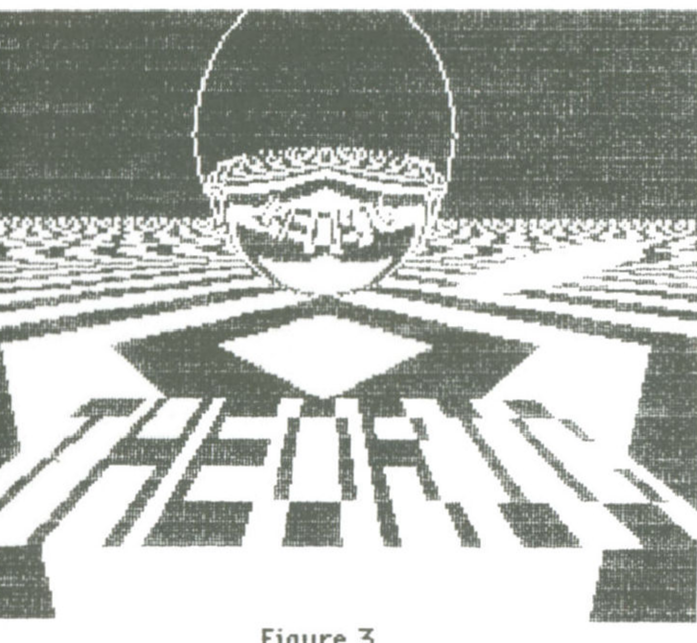


Figure 3

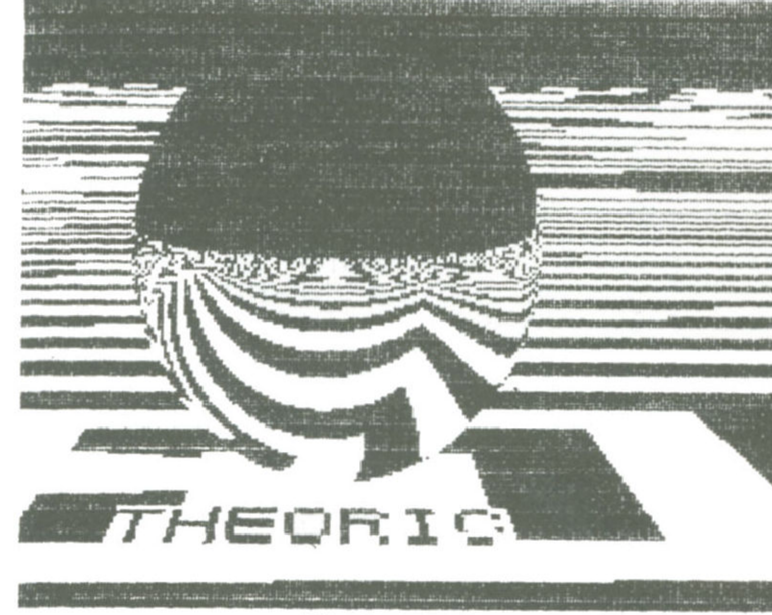


Figure 4

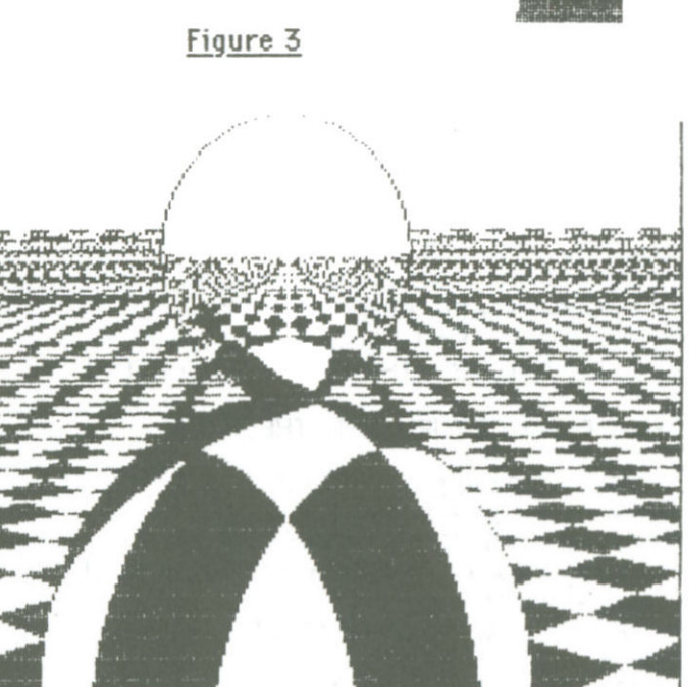


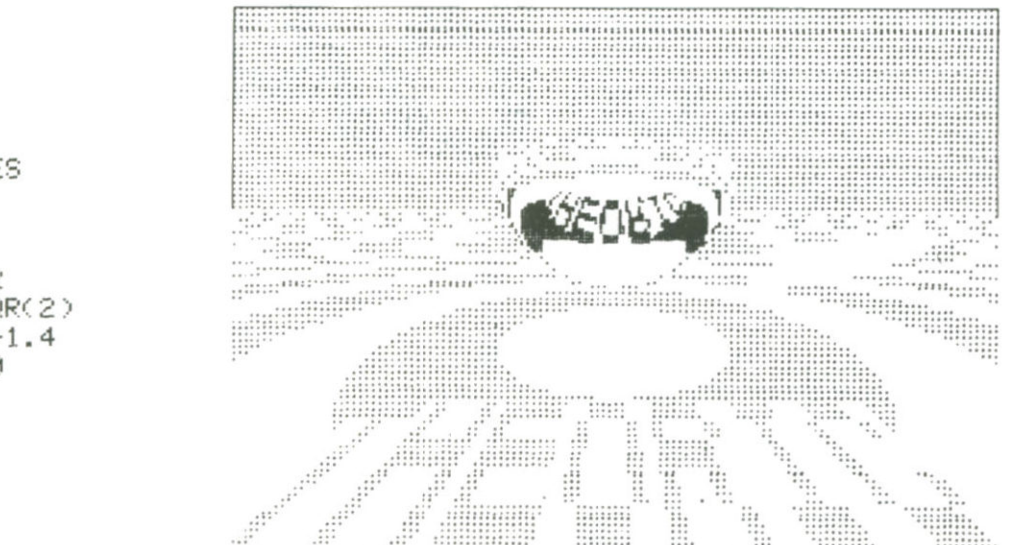
Figure 6



Figure 7

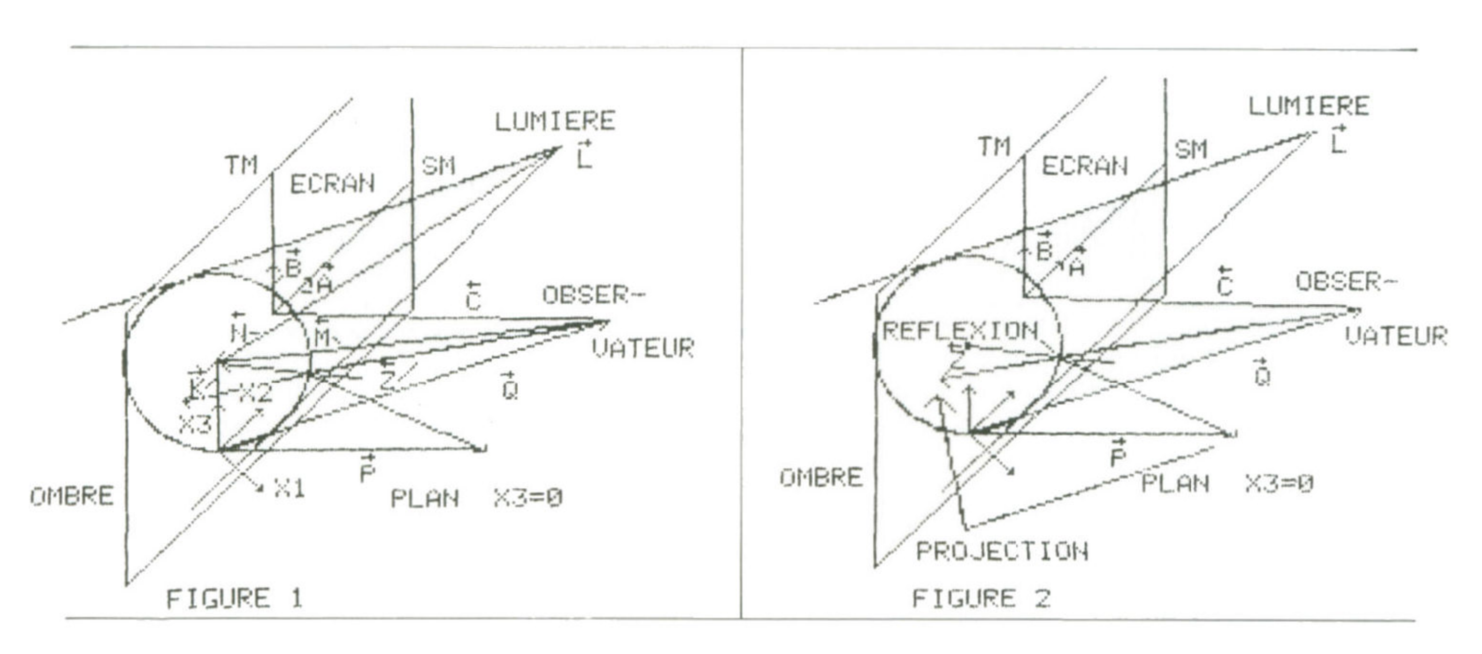
```
0 REM REFLEXION D'UN RAYON SUR UNE SPHERE DANS L'ESPACE
1 REM PROJECTION D'UN DESSIN DANS LE PLAN X3=0 SUR L'ECRAN
2 REM DEMARREZ LE PROGRAMME DANS LA SOIREE ET DORMEZ TRANQUILLEMENT...
3 REM D'APRES UNE IDEE DE MARTIN GISSER, C'T 1, 1986
10 REM INITIALISATION
15 DIMA(41,4):GOSUB1000:REM THEORIC
20 INPUT" CENTRE DE LA SPHERE (0,0,K3)=1) ";K3:K1=0:K2=0 :REM 1.41
30 INPUT"COORDONNEES DU SPECTATEUR Q3>0!"; Q1,Q2,Q3:REM 3,0,1
40 INPUT"COORDONNES DE L'ECRAN RELATIVES AU SPECTATEUR ";C1,C2,C3:REM-2,0,-.5
50 INPUT"COORDONNEES DE LA SOURCE LUMINEUSE ";L1,L2,L3:REM 35,-35,7
60 INPUT"RAPPORT LARGEUR/HAUTEUR DE L'ECRAN ";TV:REM 1.45
70 INPUT"LARGEUR DE L'ECRAN ";SM:SM=SM/2:TM=SM/TV:REM 4.7
80 INPUT " PAS DE TRAVAIL: 1<=ST<=5";ST:REM POUR UN TEST PLUS RAPIDE: ST=5
90 HIRES:DOKE#306,#FFFF
100 REM EVALUATION DES VALEURS CONSTANTES
110 CC=C1*C1+C2*C2+C3*C3:CD=SQR(CC-C3*C3)
120 A1=C2/CD:A2=-C1/CD:A3=0:B1=-C1*C3/(SQR(CC)*CD):B2=-C2*C3/(SQR(CC)*CD)
130 B3=CD/SQR(CC):N1=-L1:N2=-L2:N3=K3-L3:NN=N1*N1+N2*N2+N3*N3
140 M1=-Q1:M2=-Q2:M3=K3-Q3:MM=M1*M1+M2*M2+M3*M3
150 MA=M1*A1+M2*A2+M3*A3:MB=M1*B1+M2*B2+M3*B3:MC=M1*C1+M2*C2+M3*C3
160 AA=A1*A1+A2*A2+A3*A3:BB=B1*B1+B2*B2+B3*B3
170 CA=C1*A1+C2*A2+C3*A3:CB=C1*B1+C2*B2+C3*B3:AB=A1*B1+A2*B2+A3*B3
180 MODE=0:OLDMODE=0:REM DETECTION BORD DE LA SPHERE
200 REM DEUX BOUCLES
210 FORY=198TO0STEP-ST:T=(99-Y)*TM/99
220 MY=MC+MB*T:ZY=CC+BB*T*T+2*CB*T
230 Z4=C1+B1*T:Z5=C2+B2*T:Z6=C3+B3*T
240 FOR X= 0TO 238STEP ST: S=(X-119)*SM/119
250 MZ=MY+MA*S:ZZ=ZY+AA*S*S+2*CA*S+2*AB*S*T
260 Z1=Z4+A1*S:Z2=Z5+A2*S:Z3=Z6+A3*S
270 DI=MZ*MZ-ZZ*(MM-1)
280 IF MZ>0 AND DI>0 THEN DI=SQR(DI):MODE=1:GOSUB500:GOTO300
290 MODE=0:  PX=Q1:PY=Q2:PZ=Q3
295 IF MODE<>OLDMODE THEN OLDMODE=MODE:CURSET X-1,Y,0:REM BORD DE SPHERE
300 IF Z3=0 THEN Z3=1E-9
310 J=-PZ/Z3:IF J>=0 THEN GOSUB 400ELSE CURSET X,Y,1
320 NEXT:NEXT:DOKE#306,10000:STOP
400 REM CALCUL INTERSECTION RAYON-PLAN X3=0
410 P1=PX+J*Z1:P2=PY+J*Z2:LX=P1-L1:LY=P2-L2:LZ=-L3
420 LU=LX*N1+LY*N2+LZ*N3:LL=LX*LX+LY*LY+LZ*LZ
430 IF LU>0 AND LU*LU>=LL*(NN-1) THEN CURSET X,Y,0:RETURN:REM OMBRE
440 GOSUB 600:CURSET X,Y,FB:RETURN
499 REM SPHERE TOUCHEE-REFLEXION
500 IF MODE<>OLDMODE THEN OLDMODE=MODE:CURSET X-1,Y,0:REM BORD
510 PX=Q1+Z1*(MZ-DI)/ZZ:PY=Q2+Z2*(MZ-DI)/ZZ:PZ=Q3+Z3*(MZ-DI)/ZZ
520 ZR=Z1*PX+Z2*PY+Z3*(PZ-K3):ZK=-2*ZR
530 Z1=Z1+ZK*PX:Z2=Z2+ZK*PY:Z3=Z3+ZK*(PZ-K3)
540 RETURN
600 REM DESSIN DANS LE PLAN X3=0
610 IF P1>1 AND P1<=2 AND P2>=-1.5 AND P2<1.5 THEN 650:REM THEORIC
620 PU=ABS(P1)+ABS(P2):IF 2*INT(PU/2)=INT(PU) THEN FB=0 ELSE FB=1
625 REM OU: IF 2*INT(P1/2)-INT(P1)=2*INT(P2/2)-INT(P2) THEN FB=0 ELSE FB=1
630 RETURN
650 F=INT(14*(P2+1.5)):G=INT(5*(2-P1)) :FB=A(F,G):RETURN:REM THEORIC
999 REM MATRICE THEORIC
1000 A(0,4)=1:A(1,4)=1:F=2:GOSUB1100:A(3,4)=1:A(4,4)=1
1010 F=6:GOSUB1100:A(7,2)=1:F=10:GOSUB1100:A(8,2)=1:A(9,2)=1
1020 F=12:GOSUB1100:G=4:GOSUB1200:G=2:GOSUB1200:G=0:GOSUB1200
1030 A(18,1)=1:A(18,2)=1:A(18,3)=1:A(19,0)=1:A(19,4)=1:A(20,0)=1:A(20,4)=1
1035 A(21,0)=1:A(21,4)=1:A(22,1)=1:A(22,2)=1:A(22,3)=1
1040 F=24:GOSUB1100:A(25,2)=1:A(25,4)=1:A(26,2)=1:A(26,4)=1:A(27,1)=1
1045 A(27,2)=1:A(27,4)=1:A(28,3)=1:A(28,0)=1
1050 F=32:GOSUB 1100:A(31,4)=1:A(33,4)=1:A(31,0)=1:A(33,0)=1
1060 F=36:GOSUB 1100:A(36,4)=0:A(36,0)=0:A(37,4)=1:A(38,4)=1:A(39,4)=1
1065 A(40,3)=1:A(40,1)=1:A(39,0)=1:A(38,0)=1:A(37,0)=1
```

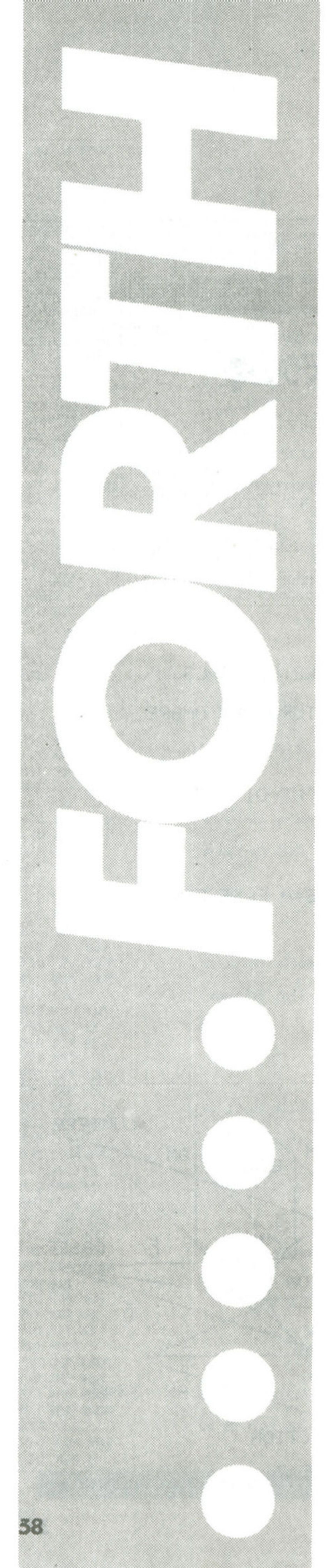
```
1070 RETURN
1100 FOR G=0TO4:A(F,G)=1:NEXT:RETURN
1200 FORF=12 TO 16:A(F,G)=1:NEXT:RETURN
```



PROGRAMME DEUX SPHERES

```
0 REM DEUX SPHERES
20 K3=3:K2=0:K1=0
30 Q1=7:Q2=-7:Q3=3.2
35 EF=(SQR(98)+1)/SQR(2)
40 C1=-EF:C2=EF:C3=-1.4
50 L1=40:L2=40:L3=40
60 TV=1.45
70 SM=3.5 :TM=SM/TV
. . . . . .
. . . . . .
. . . . . .
. . . . . .
400 REM CALCUL INTERSECTION RAYON-PLAN X3=0
405 GOSUB1000: REM 2. SPHERE
410 P1=PX+J*Z1:P2=PY+J*Z2:P3=PZ+J*Z3 :LX=P1-L1:LY=P2-L2:LZ=P3-L3
420 LU=LX*N1+LY*N2+LZ*N3:LL=LX*LX+LY*LY+LZ*LZ
430 IF LU>0 AND LU*LU>=LL*(NN-1) THEN CURSET X,Y,0:RETURN:REM OMBRE
440 GOSUB 600:CURSET X,Y,FB:RETURN
499 REM SPHERE TOUCHEE-REFLEXION
500 IF MODE<>OLDMODE THEN OLDMODE=MODE:CURSET X-1,Y,0
510 PX=Q1+Z1*(MZ-DI)/ZZ:PY=Q2+Z2*(MZ-DI)/ZZ:PZ=Q3+Z3*(MZ-DI)/ZZ
520 ZR=Z1*PX+Z2*PY+Z3*(PZ-K3):ZK=-2*ZR
530 Z1=Z1+ZK*PX:Z2=Z2+ZK*PY:Z3=Z3+ZK*(PZ-K3)
540 RETURN
600 REM IMAGE DANS LE PLAN X3=0
620 IF 2*INT(P1/2)-INT(P1)=2*INT(P2/2)-INT(P2) THEN FB=0 ELSE FB=1
630 RETURN
1000 ZZ=Z1*Z1+Z2*Z2+Z3*Z3:ZP=PX*Z1+PY*Z2+PZ*Z3
1010 PP=PX*PX+PY*PY+PZ*PZ:DY=(ZP*ZP-ZZ*(PP-4))
1020 IF DY<0 THEN RETURN:REM 2. SPHERE PAS TOUCHEE
1030 J1=-(ZP+SQR(DY))/ZZ:IF PZ+J1*Z3>0 THEN J=J1
1040 RETURN
```





BASILE

Michel ZUPAN

BASE DE DONNEES FORTH

Les écrans FORTH sont loin d'être réservés aux programmes à compiler. Voici, par exemple, comment utiliser la vaste mémoire virtuelle en fichiers avec une vraie gestion de base de données.

BASILE est un vocabulaire FORTH de bases de données gérant des fichiers ASCII à accès direct. Malgré sa taille inférieure à 2 ko, le noyau proposé ici n'a rien à envier à notre vieil ORIC-BASE et son Basic un peu essoufflé.

La puissance de BASILE viendra de ses développements : il s'agit avant tout d'un vocabulaire spécifique avec lequel vous écrirez vos propres applications adaptées à vos besoins, tant il est vrai que toute base de données se programme.

Bien sûr, BASILE est d'abord destiné aux versions disque (FORTH-JASMIN ou T-FORTH), mais les possesseurs de versions cassette peuvent très bien l'expérimenter tel quel.

BASILE est en Fig-Forth standard et donc portable sur tout système : qui a dit qu'il fallait un ORIC pour lire THEORIC ?

Je profite du fait que BASILE est auto-documenté et suivi d'un exemple d'utilisation pour répondre aux lecteurs qui nous demandent des cours d'initition au Forth.

Forth est un langage dont les joies sont à la mesure des efforts d'apprentissage. Rien ne remplace les longs essais au clavier avec un bon manuel. Ecrits par des passionnés, les manuels du langage sont souvent excellents s'ils sont parfois ardus.

Mais, voyez les autres revues : après quelques cours où l'on explique en deux pages la fonction DUP, on achève de décourager les lecteurs et la rubrique s'arrête vite là...

A notre avis, seule la publication de programmes est à même d'encourager les efforts et d'élargir le cercle des fanatiques.

Si vous débutez en Forth, laissez pour l'instant BASILE de côté. Vous y reviendrez avec plaisir quand vous saurez apprécier les mots de définition et multi-définition, le traitement des chaînes, la gestion des tampons, l'exécution vectorisée ou les tris Shell. C'est la compréhension intime du programme qui vous fera le dépasser.

Signalons enfin à tous les passionnés de FORTH, LOGO, LISP, MUMPS et autres C (mais oui : pourquoi pas sur ORIC ?) l'excellente association JEDI* et la devise de son bulletin : "Que le Forth soit avec vous !".

*Association JEDI : 8 rue Poirier de Narçay, 75014 PARIS.

```
SCRN# 71
( BASILE  1/9  : Base de données Fig-Forth  MZ.03.86 )
FORTH DEFINITIONS DECIMAL VOCABULARY BASILE IMMEDIATE
BASILE DEFINITIONS
0 VARIABLE TRAVAIL ( fichier de travail )
0 VARIABLE POINTEUR ( pointeur de la fiche courante )
0 VARIABLE #CHAMP ( pointeur de champ )
: LG-FICHE ( --- lg ) TRAVAIL @ 2+ @ ;
: NB-CHAMPS ( --- n ) TRAVAIL @ 6 + @ ;
: FIN ( --- adr ) TRAVAIL @ 4 + ;
: NB-FICHES ( --- nb ) FIN @ 1 - ;
: ALLONGE 1 FIN +! ; : DIMINUE -1 FIN +! ;
: DEJA ( n --- ) 1+ FIN ! ;
  ( permet de reprendre un fichier de n fiches déjà défini )
: POINTE ( fiche --- ) POINTEUR ! ;
```

```
: (CHAMP)  ( n --- champ ) ( donne le Nieme champ du fichier )
  NB-CHAMPS SWAP - 2 * TRAVAIL @ 8 + + @ ;  -->

SCRN# 72
( BASILE  2/9  : définitions de champs et fichiers )
: :CHAMPS ( n --- champ1..champN N lg-fiche )
  ( multi-définisseur de n champs : longueurs et noms )
  0 2DUP DO 32 WORD HERE NUMBER DROP 2DUP VARIABLE , +
  LATEST PFA ROT ROT LOOP ;
: :FICHIER ( champ1 .. champN N lg-fiche écran-debut )
  ( définition d'un fichier de N champs, fiches de lg-fiche )
  ( et débutant au numéro d'écran-debut )
  <BUILDS , , 1 , DUP , 0 DO , LOOP DOES> TRAVAIL ! ;
: :SOUS-FICHIER ( ecran --- ) ( définition d'un fichier à cet )
  ( écran de même structure que le fichier de travail )
  >R NB-CHAMPS 1+ 1 DO I (CHAMP) LOOP NB-CHAMPS
  LG-FICHE R> :FICHIER ;
: FICHE ( --- adr ) ( fournit adresse de fiche courante FC )
  POINTEUR @ B/BUF LG-FICHE / /MOD TRAVAIL @ @ B/SCR * +
  BLOCK SWAP LG-FICHE * + ;  -->

SCRN# 73
( BASILE  3/9  : impressions formatées )
: CHAMP ( champ --- adr ) ( fournit l'adresse champ FC )
  DUP 2+ @ FICHE + SWAP @ ;
: .IDCHAMP ( champ --- ) ( affiche le titre du champ )
  NFA DUP C@ 31 AND 0 DO 1+ DUP C@ 127 AND EMIT LOOP DROP ;
46 VARIABLE CAR ( caractère de remplissage )
25 VARIABLE TAB ( variable tabulations )
: CARS ( n --- ) -DUP IF 0 DO CAR @ EMIT LOOP THEN ;
  ( envoie n caractères CAR )
: RTYPE ( adr count tab --- ) ( TYPE formaté a droite )
  SWAP OVER MIN SWAP OVER - CARS TYPE ;
: LTYPE ( adr count tab --- ) ( TYPE formaté a gauche )
  SWAP OVER MIN SWAP OVER - >R TYPE R> CARS ;
: .FICHE ( --- ) ( Affiche la fiche courante )
  NB-CHAMPS 1+ 1 DO I CR (CHAMP) DUP .IDCHAMP
 CHAMP -TRAILING TAB @ OUT @ - RTYPE LOOP CR ; -->

SCRN# 74
( BASILE  4/9  : entrées de chaînes )
: (") ( compilation " ) R PAD OVER C@ 1+ DUP R> + >R CMOVE ;
: " ( --- ) 34 STATE @ IF COMPILE (") WORD HERE C@ 1+ ALLOT
  ELSE WORD HERE PAD OVER C@ 1+ CMOVE THEN ; IMMEDIATE
  ( " texte" place une chaîne explicite dans PAD )
: MET ( champ --- ) ( met chaîne dans champ de FP )
  CHAMP 2DUP BLANKS PAD COUNT ROT MIN >R SWAP R> CMOVE UPDATE ;
: ?" ( --- ) ( attente au clavier d'une chaîne pour PAD )
  BLK @ >R IN @ >R 0 BLK ! SPACE 63 EMIT SPACE QUERY
  [COMPILE] " R> IN ! R> BLK ! ;
: DEMANDE ( --- ) ( entrée au clavier champ par champ de FP )
  NB-CHAMPS 1+ 1 DO I CR (CHAMP) DUP .IDCHAMP ?" TIB @ C@ 0=
  IF LEAVE ELSE MET THEN LOOP ;
: ENTRE ( --- ) BEGIN FIN @ POINTE DEMANDE TIB @ C@
  WHILE ALLONGE CR REPEAT ; -->
  ( acquisition des fiches au clavier jusqu'à champ vide )

SCRN# 75
( BASILE  5/9  : manipulations de fiches )
: EFFACE ( --- ) FICHE LG-FICHE BLANKS UPDATE ;
  ( efface la fiche pointée )
: PREND ( --- ) FICHE PAD LG-FICHE CMOVE ;
  ( prend la fiche pointée dans PAD )
: PLACE ( --- ) PAD FICHE LG-FICHE CMOVE UPDATE ;
  ( place fiche du PAD au niveau pointé )
: COPIE ( n1 n2 --- )
  ( copie fiche n1 en n2 )
  >R POINTE PREND R> POINTE PLACE ;
-->

SCRN# 76
( BASILE  6/9  : manipulations de fiches )
: INSERRE ( --- )
  ( fait de la place pour une fiche au niveau pointé )
  POINTEUR @ DUP FIN @ DO I 1- I COPIE -1 +LOOP
  ALLONGE POINTE EFFACE ;
: DETRUIT ( --- )
  ( détruit la fiche pointée )
  POINTEUR @ FIN @ OVER DO I 1+ I COPIE LOOP DIMINUE POINTE ;
: AJOUTE ( --- )
  ( ajoute une fiche en fin de fichier de travail )
  FIN @ POINTE PLACE ALLONGE ;
: PERMUTE ( n1 n2 --- )
  ( permute les fiches n1 et n2 )
  DUP POINTE PREND OVER POINTE FICHE
  SWAP POINTE FICHE LG-FICHE CMOVE UPDATE POINTE PLACE ;
-->

SCRN# 77
( BASILE  7/9  : parcours du fichier )
  ( compteur de fiches lors de sélections )
0 VARIABLE COMPTEUR : RAZ 0 COMPTEUR ! ;
: COMPTE 1 COMPTEUR +! ;
: COMBIEN CR CR COMPTEUR @ . ." fiches selectionnees " ;
  ( procédures vectorisées sur toutes les fiches )
0 VARIABLE VECTEUR ( vecteur d'action sur une fiche )
: VAS-Y ( pfa --- )
  ( applique a toutes les fiches une procédure )
  VECTEUR ! RAZ FIN @ 1 DO I POINTE VECTEUR @ CFA EXECUTE LOOP
  COMBIEN ;
: .FICHIER ' .FICHE VAS-Y ;
  ( affiche tout le fichier de travail )
-->

SCRN# 78
( BASILE  8/9  : comparaisons de chaînes )
: TROUVE ( champ --- flag )
  ( teste présence chaîne du PAD dans le champ )
  CHAMP PAD COUNT EDITOR MATCH DROP ;
: COMPARE ( adr1 adr2 lg --- flag )
  ( compare lg octets de adr1 et adr2 : flag -1 0 ou 1 )
  ( selon que la chaîne adr1 est < = ou > a celle d'adr2 )
  1 ROT ROT OVER + 1+ SWAP DO DROP DUP C@ I C@ - -DUP
  IF 1 SWAP +- LEAVE ELSE 1+ 0 THEN LOOP SWAP DROP ;
  ( ce mot aurait intérêt à être ré-écrit avec assembler )
-->

SCRN# 79
( BASILE  9/9  : champs numériques et tris )
: VALEUR ( champ --- n )
  CHAMP DUP PAD C! PAD 1+ SWAP CMOVE PAD NUMBER DROP ;
: #MET ( n champ --- )
  ( met une valeur numérique dans un champ )
  #CHAMP ! S->D SWAP OVER DABS <# #S SIGN #> #CHAMP @
  CHAMP 2DUP BLANKS ROT MIN CMOVE UPDATE ;
( opérations de tri )
1 VARIABLE SENS : INVERSE -1 SENS ! ; : NORMAL 1 SENS ! ;
  ( déterminent le sens des tris ultérieurs )
: TRIE ( champ --- )
  ( trie fichier de travail selon champ donné et sens fixé )
#CHAMP ! NB-FICHES BEGIN 2 / -DUP WHILE NB-FICHES 1+ OVER 1+ DO
I OVER - BEGIN 2DUP + 2DUP POINTE #CHAMP @ CHAMP PAD SWAP CMOVE
POINTE #CHAMP @ CHAMP PAD SWAP COMPARE SENS @ = IF OVER PERMUTE
OVER - DUP 1 < ELSE DROP 1 THEN UNTIL DROP LOOP REPEAT ;  ;S

SCRN# 100
( Fichier BIBLIORIC   15 fiches mise a jour : 29/3/86 )
Forth pour Oric               ASN                SORICOM    13
Conduite de l'Oric-1          Astier             EYROLLES   14
Guide de l'Oric               Bayvejiel          EDIMICRO   11
Oric-1 pour tous              Boisgontier        PSI        13
Manuel de reference           Cheniere           ISOSOFT    18
Clefs pour l'Oric             Flesselles         PSI        15
Extensions pour Oric          Garrec Vignet      EYROLLES   15
Pilotez votre Oric            Gueulle            ETSF       14
Assembleur de l'Oric          Henrot             PSI        15
Interfaces pour Oric-Atmos    Levrel             SORICOM    16
Tours de Forth                Petreman Rousseau  EYROLLES   17
Tout savoir sur Oric          Politis Vanryb     EYROLLES   14
Forth                         Salman et Coll.    EYROLLES   17
60 solutions pour Oric        Schulz             ETSF       15
Programmer le Forth           Van Loo            MARABOUT   16

SCRN# 80
( BASILE : exemple de création de fichier )
BASILE DEFINITIONS DECIMAL
4 :CHAMPS 30 TITRE 20 AUTEUR 10 EDITEUR 4 NOTE
100 :FICHIER BIBLIORIC
BIBLIORIC ENTRE ;S

Cet écran définit un  fichier  BIBLIORIC  débutant  en  # 100
composé de fiches de 4 champs TITRE AUTEUR  EDITEUR  et  NOTE
de 30 20 10 et 4 caractères.
La longueur de fiche à 64 octets a été choisie comme diviseur
entier des buffers pour optimiser l'occupation de la mémoire.
Reportez-vous à l'écran # 100 pour entrer  les  15  premières
fiches de cet exemple.

SCRN# 81
( BASILE : exemple d'utilisation )
BASILE DEFINITIONS BIBLIORIC
110 :SOUS-FICHIER WOUAHH
NORMAL TITRE TRIE
: SESSION
" SORICOM" EDITEUR TROUVE IF " SORACOM" EDITEUR MET THEN
" Forth" TITRE TROUVE   NOTE VALEUR 15 >   OR
IF COMPTE CR TITRE CHAMP TYPE
   PREND WOUAHH AJOUTE BIBLIORIC THEN ;
' SESSION VAS-Y FORGET SESSION ;S

Cet écran trie BIBLIORIC selon les titres des  ouvrages  puis
sélectionne les fiches dont le titre contient 'Forth' ou dont
la note est supérieure a 15. Ces fiches  sont  comptées,  les
titres affichés et les fiches copiées dans un fichier  WOUAHH
en #110. Au passage SORICOM est remplacé par SORACOM...
```

MINI INTERPRETEUR

Pierre CHICOURRAT

Dans un numéro précédent, THEORIC vous avait proposé un exemple de routine "Mini-Interpréteur" permettant d'utiliser le "!" pour ajouter des mots au Basic de l'ORIC. Voici, aujourd'hui, un exemple d'utilisation de cette routine.

Le plus long sera pour vous de saisir le programme Basic, se chargeant d'implanter la routine LM et de la reloger à l'adresse que vous voulez (suivant la place libre en mémoire). Une fois implantée, vous pourrez, grâce à cette routine, utiliser quelques mots nouveaux dans vos programmes Basic, tous accessibles par "!" (c'est-à-dire que leur nom devra être précédé du !). Passons en revue ces nouvelles commandes.

Tout d'abord, cette routine améliorera quelques mots du Basic normal : GOTO, GOSUB, REPEAT et les mots associés : RETURN, POP, UNTIL, PULL. L'instruction !GOTO fonctionnera exactement comme celle du Basic, mis à part que vous pourrez également faire un saut sur une étiquette alphanumérique et non plus nécessairement sur un numéro de ligne. Un exemple étant plus parlant :

10 !GOTO "exemple" (GOTO a l'étiquette "exemple)
20 ...
99 REM "exemple" (étiquette exemple sur laquelle se fait le GOTO)
100 PRINT "OK: ÇA,MARCHE"

Bien sûr, le paramètre du GOTO peut aussi être une variable alphanumérique. Par exemple :

30 A$ = "SPRG2"
40 !GOTO A$ (GOTO sur l'étiquette contenu dans A)
50 ...
200 REM "SPRG2" (étiquette pour le GOTO - dans un REM entre guillemets)
210 PRINT "Le GOTO a fonctionné"

On peut, dans un programme, utiliser autant d'étiquettes que l'on désire (le nombre n'est absolument pas limité par la place mémoire). Par ce procédé, vous vous affranchissez des numéros de ligne ; le programme devient beaucoup plus parlant et, de plus, la rénumérotation des lignes ne nécessite plus qu'une routine extrêmement simplifiée (puisque les GOTO n'ont plus à être adaptés).

Le GOSUB subit également la même amélioration, mais ce n'est pas la seule. Vous pourrez désormais utiliser autant de GOSUB emboîtés que vous le désirez. Prenons, par exemple, le petit programme suivant :

10 I = I + 1:GOSUB 10

Faites RUN. Peu de temps après, vous obtenez un message d'erreur "OUT OF MEMORY" ; il y a trop de GOSUB emboîtés. Si vous lisez la valeur atteinte par la variable I, vous trouverez quelque chose comme 25. Cette limite du Basic vous empêche de faire des programmes en tant soit peu récursifs. Par contre, avec le nouveau GOSUB, vous pourrez utiliser beaucoup plus de sous-niveaux de sous-programmes. Ainsi, le programme :

10 I = I + 1:PRINT I:!GOSUB 10

ne se plantera pas pour I = 500, ni même après, suivant la place que vous aurez réservée à la pile des retours ! En effet, à chaque fois que l'interpréteur effectue un GOSUB (d'ailleurs, la même chose se passe pour un REPEAT), il stocke dans la pile du microprocesseur (située entre # 100 et # 1FF) l'adresse du caractère qui suit le GOSUB, le numéro de la ligne où se fait le GOSUB et enfin un code permettant de savoir à quoi correspond tout ce qui a été sauvegardé dans la pile (le code de l'instruction GOSUB ou REPEAT). Ainsi donc, chaque GOSUB occupe 5 octets plus deux octets d'adresse de retour pour le microprocesseur, soit 7 octets occupés. Comme la pile est de longueur très limitée (elle n'est pas entièrement consacrée à servir de pile de retour), ceci explique que l'on ne puisse pas emboîter autant de GOSUB que l'on veut. L'astuce consiste donc à utiliser une autre pile des retours, que l'on loge ailleurs en mémoire et dont la longueur sera plus importante (puisque l'on ne sera plus limité). De plus, au lieu de sauvegarder 7 valeurs, on n'en sauvegarde plus que 3, l'adresse du caractère qui suit le GOSUB et le code de l'instruction, le numéro de ligne est retrouvé au moment où l'on fait le RETURN et l'adresse de retour pour le microprocesseur n'est plus utile puisque la pile normale n'est pas affectée par le GOSUB. Ceci oblige également à réécrire le RETURN et le POP afin qu'ils soient capables d'utiliser la nouvelle pile des retours.

Le REPEAT subit exactement la même modification, ainsi que les deux mots associés à cette instruction : PULL et UNTIL. Ces instructions utiliseront la même pile que le nouveau GOSUB et donc le nombre de REPEAT que vous pourrez emboîter dépendra de la

longueur choisie pour cette pile. Le programme d'implantation du L.M. vous permettra de choisir l'emplacement de cette pile et, par voie de conséquence, de fixer sa longueur (à titre indicatif, une pile de 99 octets permettra environ 33 niveaux de sous-programmes, si vous en voulez plus, rien ne vous empêche de réserver une pile plus longue ; on peut sans problème utiliser 1000 niveaux de sous-programmes !).

Passons maintenant aux nouvelles commandes créées par cette routine. Elles sont toutes utilisables dans vos programmes Basic, à condition de faire précéder leur nom, comme indiqué par "!".

!WHILE et !WEND : Structure de contrôle destinée à compléter le REPEAT UNTIL. Sa syntaxe :

!WHILE condition... !WEND

Cette instruction exécute ce qui est entre le WHILE et WEND tant que la condition est vraie. Utilise la même pile que le nouveau GOSUB ; on peut donc emboîter autant de WHILE/WEND que l'on désire.
Exemple :

10 I=0
20 !WHILE I<10:I=I+1
30 PRINT I
40 !WEND

Ce programme tournera jusqu'à ce que I atteigne la valeur 10 ; à ce moment là, le WHILE fera sortir de la boucle.

!HOME : ramène le curseur en haut de l'écran mais sans effacer ce qui est affiché.

!RAND : Commande "RANDOMIZE". Permet de rendre le générateur de nombres aléatoires vraiment aléatoire en initialisant la "semence" utilisée par celui-ci. Ainsi, quand on commence un jeu de hasard, les nombres tirés n'arriveront pas toujours dans le même ordre.

!FIND [chaîne de caractères] : Outil d'aide à la programmation. Permet de retrouver dans un programme Basic les occurences de la chaîne mise entre crochets. Attention : cette chaîne ne peut pas être une variable alphanumérique ni une expression de chaînes (somme). Permet de retrouver toutes les occurences d'une commande du Basic dans un programme.
Exemple :

!FIND [GOTO]
!FIND [REM]
!FIND [Tapez]

!SWAP Variable1, Variable2 : Echange le contenu de la Variable1 et de la Variable2 de manière instantanée. Bien sûr, les deux variables doivent être du même type.
Exemple :

A=20:B=30:!SWAP A,B:PRINT A;B Affiche "30 20"
A$="Monsieur":B$="Cher":!SWAP A$,B$:PRINT A$,B$
A(0)=34:A(2)=67:!SWAP A(0), A(2):PRINT A(0);A(2)

Cette instruction opère sur tout type de variable (y compris les variables entières).

INSTRUCTIONS DE MANIPULATION DE L'AFFICHAGE

!SAVE A,X,Y;B : Sauve la partie de l'écran commençant à l'adresse A, s'étendant sur X caractères en abscisse et Y caractères en ordonnées. Met cette zone sauvegardée en mémoire à partir de l'adresse B. Comme il n'y a pas de contrôle, vous pouvez vous servir de cette instruction pour sauvegarder une zone mémoire. Cette commande permet de faire de petits dessins animés (on sauve les images et on les réaffiche à l'endroit que l'on veut).
Exemple :

!SAVE 48000,10,10;36864 : sauve le petit carré de 10 × 10 en haut à gauche de l'écran TEXT.

!LOAD A;B : Recharge à partir de l'adresse B la zone écran qui a été sauvegardée à partir de l'adresse A (les dimensions de la zone sauvegardée par !SAVE sont aussi sauvées). Cette instruction permet de faire de petits dessins animés (on peut sauvegarder et recharger une partie de l'écran TEXT ou HIRES).

Cette routine ne constitue qu'un petit exemple d'application. En fait, la commande "!" s'avère très souple pour adapter le Basic de l'ORIC à ses besoins et en faire ainsi un Basic personnalisé et vraiment efficace.

```
0 REM===================================
1 REM=                                 =
2 REM= EXEMPLE D'EXTENSION DU BASIC    =
3 REM=                                 =
4 REM=                                 =
5 REM=          (C) Pierre CHICOURRAT  =
6 REM=                                 =
7 REM===================================
10 TEXT:PAPER0:INK2:CLS:PRINTCHR$(4):PRI
NTSPC(5)CHR$(27)"A"CHR$(27)"J";
20 PRINT"EXTENSIONS AU BASIC"CHR$(4)
30 PRINT:PRINT:PRINT:PRINTSPC(16)CHR$(27
)"E"CHR$(96)" Pierre CHICOURRAT"
40 PRINT:PRINT:PRINT"A QUELLE ADRESSE VO
ULEZ-VOUS IMPLANTER"
```



```
50 PRINT"LE PROGRAMME ?":PRINT"IL FAUT Q
UE CETTE ZONE SOIT PROTEGEE"
60 INPUTAD:RESTORE:FORJ=0TO2:READU:I=AD+
U:REPEAT:READU$:U=VAL("#"+U$)
70 POKEI,U:I=I+1:UNTILU$="*":NEXTJ:PRINT
:PRINT
80 REPEAT:READU$,V$:U=VAL("#"+U$)+AD:V=V
AL("#"+V$)+AD:IFV$<>"*"THENDOKEU,V
90 UNTILU$="*"
100 REPEAT:READU$,V$,W$:IFPEEK(#FFFE)=40
THENV$=W$
110 U=VAL("#"+U$)+AD:V=VAL("#"+V$):IFU$<
>"*"THENDOKEU,V
120 UNTILU$="*":DOKE0,AD+#32F:POKEAD+#2C
2,PEEK(0):POKEAD+#2C4,PEEK(1)
130 DOKE0,AD+#586:POKEAD+6,PEEK(0):POKEA
```

```
D+8,PEEK(1)
140 DOKE0,AD+#31C:POKEAD+#296,PEEK(0):PO
KEAD+#298,PEEK(1)
150 T=AD+#546:REPEAT:READU$:U=VAL("#"+U$
)+AD:DOKE0,U:POKET,PEEK(0)
160 POKET+32,PEEK(1):T=T+1:UNTILU$="*"
170 FORI=AD+#3DETOAD+#42F:POKEI,0:NEXT
180 PRINT"QUELLE LONGUEUR VOULEZ-VOUS RE
SERVER"
190 PRINT"POUR LA "CHR$(34)"PILE DES RET
OURS"CHR$(34)
200 PRINT"POUR LES GOSUB,REPEAT, .. etc
.."
210 INPUT"(99 OCTETS = 33 NIVEAUX DE SOU
S-PROG ";L
220 I=AD+#5D0:PRINT"CETTE PILE EST DONC
IMPLANTEE DE :":PRINTHEX$(I)" A "HEX$(I+
L)
230 DOKEAD+#316,I+L:DOKEAD+#318,I:DOKEAD
+#31A,I+L:PRINT:PRINT
240 DOKE#2F5,AD:PRINT"OK: VOICI LES NOUV
ELLES INSTRUCTIONS"
250 PRINT"UTILISABLES :"
260 PRINT"!GOTO N,!GOTO "CHR$(34)"Etique
tte"CHR$(34)
270 PRINT"!GOSUB N, !GOSUB "CHR$(34)"Eti
quette"CHR$(34)
280 PRINT"!RETURN, !POP"
290 PRINT"!REPEAT, !UNTIL, !PULL":PRINT"
!WHILE Condition .... !WEND"
300 PRINT"!CLEAR :VIDE TOTALEMENT LA PIL
E DES":PRINTSPC(9)"RETOURS"
310 PRINT"!RAND":PRINT"!HOME":PRINT"!FIN
D [Chaine]":PRINT"!SWAP Variable1,Variab
le2"
320 PRINT"!SAVE A,X,Y;B :SAUVEGARDE ZONE
 ECRAN"
330 PRINT"!LOAD A;B  :AFFICHAGE D'UNE ZO
NE":PRINTSPC(14)"SAUVEGARDEE"
340 PRINT:PRINT"Tapez une touche pour co
ntinuer":POKE#2DF,0:GETR$
350 CLS:PRINT"VOULEZ-VOUS SAUVEGARDER LA
 ROUTINE"
360 PRINT"EN L.M ?  [O]/[N] ?"
370 POKE#2DF,0:GETR$:PRINTR$;:IFR$="N"TH
ENPRINT:END
380 IFR$<>"O"THENPRINTCHR$(127);:GOTO370
390 PRINT:PRINT"A QUELLE VITESSE VOULEZ-
VOUS FAIRE"
400 PRINT"LA SAUVEGARDE ?  [F]/[S] ?"
410 POKE#2DF,0:GETR$:IFR$<>"F"ANDR$<>"S"
THENPING:GOTO410
420 PRINTR$:PRINT:PRINT"METTEZ LE MAGNET
OPHONE EN POSITION"
430 PRINT"D'ENREGISTREMENT ET APPUYER SU
R UNE"
440 PRINT"TOUCHE.":POKE#2DF,0:GETA$
450 POKE770,7:FORI=0TO3000:NEXT
460 IFR$="F"THENCSAVE"EXTENSIONS",AAD,EA
D+#5BD:GOTO480
470 CSAVE"EXTENSIONS",AAD,EAD+#5BD,S
480 PRINT:PRINT"AU REDEMARRAGE; IL FAUDR
A INITIALISER"
490 PRINT"LE VECTEUR DU '!' EN FAISANT :
"
500 PRINT"  "CHR$(27)"A"CHR$(27)"TDOKE#2
F5,"HEX$(AD):PRINT
510 PRINT"APRES, VOUS POURREZ UTILISER L
ES":PRINT"NOUVELLES FONCTIONS NORMALEMEN
T."
520 PRINT:END
900 REM==================================
910 REM= DATAS DE LA ROUTINE            =
920 REM==================================
1000 DATA0
1010 DATA20,60,60,A2,00,A9,FF,A0,FF,85,0
0,84,01,A5,E9,48,A5,EA,48,A0
1020 DATA00,B1,00,D1,E9,D0,1F,E6,00,D0,0
2,E6,01,E6,E9,D0,02,E6,EA,B1
1030 DATA00,D0,EC,BD,46,65,85,00,BD,66,6
5,85,01,68,68,6C,00,00,E8,E6
1040 DATA00,D0,02,E6,01,B1,00,D0,F6,E6,0
0,D0,02,E6,01,68,A8,68,84,EA
1050 DATA85,E9,48,98,48,A0,00,B1,00,D0,B
C,A2,10,4C,85,C4,A0,00,A9,20
1060 DATAD1,E9,F0,01,60,E6,E9,D0,F3,E6,E
A,D0,EF,A5,9A,A4,9B,85,CE,84
1070 DATACF,A0,01,B1,CE,F0,20,A0,04,B1,C
E,F0,07,C9,9D,F0,19,C8,D0,F5
1080 DATAA0,00,B1,CE,48,C8,B1,CE,85,CF,6
8,85,CE,D0,E4,A5,CF,D0,E0,4C
1090 DATAF1,C9,C8,B1,CE,F0,E5,C9,22,F0,0
5,C8,D0,F5,F0,DC,C8,A2,00,B1
1100 DATACE,F0,11,C9,22,F0,0D,C9,3A,F0,0
9,DD,B5,DA,D0,C8,C8,E8,D0,EB
1110 DATAE4,02,D0,D3,60,20,8B,CE,A5,28,D
0,20,20,7A,CE,20,A0,E7,A5,9A
1120 DATAA6,9B,20,E8,C6,B0,03,4C,F1,C9,3
8,A5,CE,E9,01,85,E9,A5,CF,E9
1130 DATA00,85,EA,60,20,15,D7,85,02,A5,9
1,A4,92,8D,C0,60,8C,C1,60,20
1140 DATA71,60,A9,00,F0,DC,A0,00,48,AD,1
A,63,85,08,AD,1B,63,85,09,68
1150 DATA60,20,0A,61,91,08,AC,1A,63,D0,0
3,CE,1B,63,CE,1A,63,60,EE,1A
1160 DATA63,D0,03,EE,1B,63,20,0A,61,B1,0
8,60,20,2A,61,4C,1E,61,85,08
1170 DATAAD,1A,63,38,E5,08,85,08,AD,1B,6
3,E9,00,CD,19,63,B0,0C,F0,03
1180 DATA4C,83,C4,A5,08,CD,18,63,90,F6,6
0,48,A9,03,20,3E,61,A5,EA,20
1190 DATA19,61,A5,E9,20,19,61,68,4C,19,6
1,20,2A,61,85,E9,20,2A,61,85
1200 DATAEA,A5,E9,A4,EA,85,08,84,09,A0,0
0,B1,08,F0,0C,A5,08,D0,02,C6
1210 DATA09,C6,08,A9,00,F0,F0,A5,08,D0,0
2,C6,09,C6,08,B1,08,F0,F4,A0
1220 DATA04,B1,08,85,A8,C8,B1,08,85,A9,6
0,A9,9B,20,5F,61,20,CD,60,68
1230 DATA68,4C,AD,C8,A9,88,20,5F,61,68,6
8,4C,AD,C8,20,38,61,C9,9B,F0
1240 DATA05,A2,16,4C,85,C4,20,2A,61,E0,0
C,F0,11,20,73,61,20,1C,CA,98
1250 DATA18,65,E9,85,E9,30,02,E6,EA,60,2
0,2A,61,4C,2A,61,A2,00,4C,C6
1260 DATA61,A2,0C,4C,C6,61,20,38,61,C9,8
B,F0,05,A2,F5,4C,85,C4,E0,10
1270 DATAD0,05,84,D0,98,D0,06,20,E8,00,2
0,8B,CE,20,2A,61,A5,D0,F0,07
1280 DATA20,2A,61,20,2A,61,60,20,73,61,4
```

```
C,BC,61,A2,10,4C,FA,61,A2,00
1290 DATA4C,FA,61,AD,16,63,AC,17,63,8C,1
A,63,8C,1B,63,60,A2,01,A0,00
1300 DATAB1,CE,C9,C0,D0,3A,20,B3,62,B1,C
E,C9,20,F0,F7,B1,CE,D9,0F,63
1310 DATAD0,17,C8,C0,05,D0,F4,E8,20,A1,6
2,98,18,65,CE,85,CE,A5,CF,69
1320 DATA00,85,CF,D0,01,A0,00,B1,CE:D9,1
4,63,D0,0A,C8,C0,02,D0,F4,CA
1330 DATA8A,D0,DD,60,20,B3,62,A5,CF,C5,9
D,90,B5,D0,06,A5,CE,C5,9C,90
1340 DATAAD,A9,FF,A0,FF,20,ED,CB,68,68,4
C,A6,C4,B1,CE,D0,03,4C,B3,62
1350 DATAC9,3A,F0,F9,20,B3,62,A9,00,F0,E
E,E6,CE,D0,02,E6,CF,60,20,38
1360 DATA61,C9,F7,F0,0C,A9,FF,A0,FF,20,E
D,CB,68,68,4C,A6,C4,20,2A,61
1370 DATA20,73,61,A9,F7,20,5F,61,20,E8,0
0,20,8B,CE,A5,D0,F0,01,60,20
1380 DATA2A,61,20,2A,61,20,2A,61,A5,E9,A
4,EA,85,CE,84,CF;20,40,62,A5
1390 DATACE,A4,CF,85,E9,84,EA,20,1C,CA,9
8,18,65,E9,85,E9,A5,EA,69,00
1400 DATA85,EA,60,57,48,49,4C,45,57,80,0
0,00,00,00,00,00,57,48,49,4C
1410 DATA45,20,57,49,54,48,4F,55,54,20,5
7,45,4E,44,00,57,45,4E,44,20
1420 DATA57,49,54,48,4F,55,54,20,57,48,4
9,4C,45,00,20,E8,00,C9,5B,F0
1430 DATA05,A2,10,4C,85,C4,E6,E9,D0,02,E
6,EA,A0,00,B1,E9,F0,EF,C9,5D
1440 DATAF0,08,99,DE,63,C8,A9,00,F0,F0,A
9,00,99,DE,63,C8,98,18,65,E9
1450 DATA85,E9,90,02,E6,EA,A5,9A,A4,9B,8
5,00,84,01,A0,04,A2,00,B1,00
1460 DATAF0,27,DD,DE,63,D0,1F,C8,E8,BD,D
E,63,D0,F0,98,48,A0,02,B1,00
1470 DATAAA,C8,B1,00,20,C1,E0,A9,20,20,1
2,CC,68,A8,A9,00,F0,D6,C8,D0
1480 DATAD3,BD,DE,63,D0,10,A0,02,B1,00,A
A,C8,B1,00,20,C1,E0,A9,20,20
1490 DATAC1,E0,A0,00,B1,00,48,C8,B1,00,8
5,01,68,85,00,A5,01,D0,AB,A9
1500 DATA0D,20,12,CC,A9,0A,20,12,CC,60
1510 DATA*,#430
1520 DATAAD,77,02,AC,76,02,85,FB,84,FC,A
D,03,03,AC,04,03,85,FD,84,FE
1530 DATA60,A9,1E,4C,12,CC,20,FC,D0,85,B
8,84,B9,60,20,4A,64,85,00,84
1540 DATA01,A5,28,A4,29,85,02,84,03,20,D
9,CF,20,4A,64,A5,28,C5,02,F0
1550 DATA05,A2,A8,4C,85,C4,A5,29,C5,03,D
0,F5,A5,28,F0,04,A0,02,D0,0A
1560 DATAA5,29,F0,04,A0,01,D0,02,A0,04,B
1,30,48,B1,B8,91,00,68,91,B8
1570 DATA88,C0,FF,D0,F1,60,20,9D,E7,A5,3
3,A4,34,85,00,84,01,20,D9,CF
1580 DATA20,0C,D8,86,02,20,D9,CF,20,0C,D
8,86,03,A9,3E,20,DB,CF,20,9D
1590 DATAE7,A0,00,A5,02,91,33,C8,A5,03,9
1,33,A5,33,18,69,02,85,33,A5
1600 DATA34,69,00,85,34,A0,00,A2,00,B1,0
0,81,33,E6,33,D0,02,E6,34,C8
1610 DATAC4,02,D0,F1,A5,00,18,69,28,85,0
0,A5,01,69,00,85,01,C6,03,D0
1620 DATADC,60,20,9D,E7,A5,33,A4,34,85,0
0,84,01,A9,3E,20,D6,CF,20,9D
1630 DATAE7,A0,00,B1,00,85,02,C8,B1,00,8
5,03,A5,00,18,69,02,85,00,90
1640 DATA02,E6,01,A2,00,A0,00,A1,00,91,3
3,E6,00,D0,02,E6,01,C8,C4,02
1650 DATAD0,F1,A5,33,18,69,28,85,33,90,0
2,E6,34,C6,03,D0,DE,60
1660 DATA*,#586
1670 DATA97,00,9B,00,86,00,9C,00,8B,00,8
C,00,88,00,57,48,49,4C,45,00
1680 DATA57,80,00,48,4F,4D,45,00,52,D1,0
0,46,49,4E,44,00,53,57,41,50
1690 DATA00,53,41,56,45,00,4C,4F,41,44,0
0,BD,00,00,00
1700 DATA*
1704 REM================================
1705 REM= POUR RELOGER LA ROUTINE      =
1706 REM================================
1710 DATA1,60,2C,546,31,566,FE,C0,101,C1
,104,71,10E,31A,113,31B,11A,10A
1720 DATA11F,31A,124,31B,127,31A,12B,31A
,130,31B,133,10A,139,12A,13C,11E
1730 DATA141,31A,149,31B,14E,319,15A,318
,163,13E,168,119,16D,119,171,119
1740 DATA174,12A,179,12A,1B2,15F,1B5,CD,
1BF,15F,1C7,138,1D3,12A,1DA,173
1750 DATA1EB,12A,1EE,12A,1F3,1C6,1F8,1C6
,1FB,138,216,12A,21D,12A,220,12A
1760 DATA224,173,227,1BC,22C,1FA,231,1FA
,234,316,237,317,23A,31A,23D,31B
1770 DATA24B,2B3,256,30F,261,2A1,276,314
,285,2B3,2A6,2B3,2AD,2B3,2BB,138
1780 DATA2CE,12A,2D1,173,2D6,15F,2E4,12A
,2E7,12A,2EA,12A,2F5,240
1790 DATA35F,3DE,369,3DE,387,3DE,38E,3DE
,3AE,3DE,453,44A,465,44A,*,*
1794 REM================================
1795 REM= DATAS D'ADAPTATION POUR      =
1796 REM=    L'ORIC-ATMOS              =
1797 REM================================
1800 DATA5E,C47E,C485,A0,CA23,C9F1,CE,CF
17,CE8B,D5,CF06,CE7A,D8,E856,E7A0
1810 DATADF,C6BD,C6E8,E4,CA23,C9F1,F5,D7
D0,D715,155,C47C,C483,1BA,C8C1,C8AD
1820 DATA1C4,C8C1,C8AD,1D0,C47E,C485,1DD
,CA4E,CA1C,204,C47E,C485
1830 DATA213,CF17,CE8B,29A,CCB0,CBED,2CB
,C499,C4A6,2DC,CF17,CE8B
1840 DATA29F,C499,C4A6,2C6,CCB0,CBED,34C
,C47E,C485,39D,E0C5,E0C1
1850 DATA3BB,E0C5,E0C1,3C0,E0C5,E0C1,3D6
,CCD9,CC12,3DB,CCD9,CC12
1860 DATA44B,D188,D0FC,462,D065,CFD9,470
,C47E,C485,49B,E853,E79D
1870 DATA4A9,D8C8,D80D,4AE,D065,CFD9,4B1
,D8C8,D80D,4B8,D067,CFDB
1880 DATA300,CA4E,CA1C,3A2,CCD9,CC12,448
,CCD9,CC12,4A6,D065,CFD9
1890 DATA4FB,E853,E79D,508,D067,CFDB,50B
,E853,E79D,4BB,E853,E79D,*,*,*
1900 REM================================
1910 REM=  ADRESSES DES ROUTINES       =
1920 REM=   DEFINISSANT LES FONCTIONS =
1930 REM================================
1340 DATACD,1AF,1F5,1F0,1BC,22E,229,2D3,
2BA,445,430,347,452,49A,4FA,233,*
```

LE CLASSEMENT INSTANTANE

D. VASILJEVIC

Vous avez certainement étudié toutes les méthodes de tri, et vous voilà devant votre clavier, patientant que les interminables boucles imbriquées classent quelques malheureux tableaux ou fiches.
L'auteur de ces lignes vous invite à oublier tout cela car, à moins d'avoir à classer des nombres astronomiques à la neuvième décimale près, en pratique, vous n'en avez nullement besoin.

Que trie-t-on dans la vie courante ? La plupart du temps, on trie des nombres entiers représentant une quantité ou une propriété quelconque.
Pourtant, ces nombres étant naturels, ils occupent une place naturelle dans leur ensemble.

C'est l'idée de base : il suffit de remplacer le doute que comporte chaque tri par la certitude du classement naturel, en substituant la référence à la valeur.

Tapez les quelques lignes du listing n° 1 et lancez le programme. Les nombres entiers aléatoires, de 0 à 100, sont imprimés dans l'ordre de leur apparition et classés instantanément dans le tableau A à l'indice égal à leur valeur (ligne 10). C'est tout, le tri est terminé !
Il suffit maintenant de les imprimer (ou de les empiler dans le même tableau, la libération des places étant plus rapide que l'occupation de fait des trous).

En prenant l'indice de boucle croissant, l'ordre sera croissant et inversement. Cette fois, c'est la valeur qui est égale à l'indice et le contenu du tableau, contrairement à tous les autres types de tri, ne contient pas la valeur, mais le nombre de valeur identique, ou le zéro en l'absence d'une valeur. Ainsi, dans l'exemple n° 1, le zéro apparaît deux fois, le quatre une fois, alors que 1, 2 et 3 n'existent pas, etc.

En pratique, il ne suffit pas de classer les valeurs, il faut savoir, aussi à quoi elles correspondent. Un marchand de chaussures, par exemple, ayant une centaine de modèles, veut savoir combien de paires il lui reste de chaque modèle et ceci dans l'ordre croissant afin de voir quels sont les modèles qui arrivent à épuisement.
Modifier le programme selon le listing n° 2.
Cette fois, les références (numéros des modèles) seront classées en même temps que les valeurs et à l'indice = valeur.
A l'impression, les valeurs sont accompagnées de leur référence (exemple n° 2).
Pour les nombres entiers, il suffit de connaître la plus grande valeur possible, afin de dimensionner le tableau en conséquence, mais on peut, aussi, se passer de tableaux (listing n° 3). Toutefois, en Basic, cette méthode est moins rapide que celle du listing n° 1 : pour 4000 nombres 1 minute, 21 secondes

```
                Listing no  1

1 CALL#EE1A
10 DIMA(100):FORI=0TO100:A=INT(RND(1)*100):PRINTA;:A(A)=A(A)+1:NEXT:PRINT:PRINT
20 REMARQUE: Les nombres sont classes. Impression:"
40 FORI=0TO100:IFA(I)=0THEN60
50 FORJ=1TOA(I):PRINTI;:NEXTJ
60 NEXTI:PRINT:PRINT:CALL#EDE0:END
```

```
                Listing no  2

1 CALL#EE1A:DIMN$(100,100):PRINT"TYPE     NOMBRE":PRINT
10 DIMA(100):FORI=0TO100:A=INT(RND(1)*100):A(A)=A(A)+1:PRINTI,A
15 N$(A,A(A))=STR$(I):NEXT:PRINT:PRINT
20 REMARQUE: Les nombres sont classes. Impression:"
25 PRINT"NOMBRE    TYPE":PRINT
40 FORI=0TO100:IFA(I)=0THEN60
50 FORJ=1TOA(I):PRINTI,N$(I,J):NEXTJ
60 NEXTI:PRINT:PRINT:CALL#EDE0:END
```

```
                Listing no  3

1 CALL#EE1A:CLS:PRINT"PATIENTEZ":FORI=#2000TO#2000+2*4000:POKEI,0:NEXT
10 FORI=0TO4000:A=INT(RND(1)*4000):DOKE#2000+2*A,DEEK(#2000+2*A)+1:NEXT
20 REMARQUE: Les nombres sont classes. Impression:"
40 FORI=0TO4000:IFDEEK(#2000+2*I)=0THEN60
50 FORJ=1TODEEK(#2000+2*I):PRINTI;:NEXTJ
60 NEXTI:PRINT:PRINT:CALL#EDE0:END
```

```
                Listing no  4

1 CALL#EE1A
10 DIMA(1000):FORI=0TO99:A=(RND(1)*10):PRINTA;:A=INT(-A*100)*(-1)
15 A(A)=A(A)+1:NEXT:PRINT:PRINT
20 REMARQUE: Les nombres sont classes. Impression:"
40 FORI=0TO999:IFA(I)=0THEN60
50 FORJ=1TOA(I):PRINTI/100;:NEXTJ
60 NEXTI:PRINT:PRINT:CALL#EDE0:END
```

```
                Listing no  5

1 CALL#EE1A:S=1
10 DIMA(1000):FORI=0TO49:A=(RND(1)*5)*S:PRINTA;:A=INT(-A*100)*(-1):S=-S
15 A(A+500)=A(A+500)+1:NEXT:PRINT:PRINT
20 REMARQUE: Les nombres sont classes. Impression:"
40 FORI=0TO999:IFA(I)=0THEN60
50 FORJ=1TOA(I):PRINT(I-500)/100;:NEXTJ
60 NEXTI:PRINT:PRINT:CALL#EDE0:END
```

```
                Listing no  6

1 HIRES:CALL#EE1A:PRINT"TYPE     NOMBRE":PRINT
10 FORX=0TO239:Y=INT(RND(1)*199):CURSETX,Y,1:PRINTX,Y:NEXT:PRINT:PRINT
20 REMARQUE: Les nombres sont classes. Impression:"
25 PRINT"NOMBRE    TYPE":PRINT
40 FOR Y=0TO199:FORX=0TO239:CURSETX,Y,3:IFPOINT(X,Y)=0THEN60
50 PRINTY,X
60 NEXTX,Y:PRINT:PRINT:CALL#EDE0:END
```

```
                Listing no  7

1 HIRES:CALL#EE1A:PRINT"NOMBRE":PRINT
10 FORI=0TO48000:A=RND(1)*48000:A=INT(A):PRINTA
15 Y=INT(A/240):X=240*(A/240-Y):IFX-INT(X)>.999THENX=X+1
16 CURSETX,Y,1:NEXT
20 REMARQUE: Les nombres sont classes. Impression:"
25 PRINT"NOMBRE":PRINT
40 FORY=0TO199:FORX=0TO239:CURSETX,Y,3:IFPOINT(X,Y)=0THEN60
50 PRINT240*Y+X
60 NEXTX,Y:PRINT:PRINT:CALL#EDE0:END
```

pour la méthode de tableau, contre 1 minute, 42 secondes pour "Doke". Le temps à prendre en considération est celui entre "RUN" et la fin de la première impression.

On peut classer de même manière et aussi rapidement les nombres décimaux. Il faut fixer la précision nécessaire (dans la vie courante deux décimales derrière la virgule). Utilisez le listing n° 4 (exemple n° 3).

Pour les nombres négatifs, il faut diviser le tableau en deux parties égales, le zéro au milieu (listing n° 5 et exemple n° 4).

On peut conclure que ce type de classement n'est limité que par la taille de la mémoire, mais ceci est valable pour tous les programmes opérant sur tableaux.

Les personnes possédant un lecteur de disquettes peuvent concevoir des fichiers "DATA" indicés et augmenter encore les possibilités. Ce fichier sera de préférence unique et séquentiel pourvu que leur SED possède une instruction de repositionnement du pointeur.

Pour les autres, moins favorisés, voici une manière d'augmenter la capacité : le classement synoptique par substitution des coordonnées (listing n° 6).

Et pour en finir avec les listings, le n° 7, le délire : classement de 48000 valeurs comprises entre 0 et 47999 (sans références). Durée : 47 minutes, 10 secondes. Mais, soyons réalistes, ceci n'est qu'une démonstration.

Il est bien évident que tous ces programmes traduits en langage machine gagneront encore en rapidité.

On peut ainsi imaginer une toute petite routine, synthèse de tous ces petits programmes, placée dans la zone du deuxième jeu de caractères et dans la zone de réserve (BFE0-BFFF) libérant en mode texte toute la mémoire comprise entre # 400 et # B3FF. En codant la partie valeur sur chaque bit et la partie référence sur un ou deux octets selon leur nombre, on obtient deux zones :

Intervalle/8 + nombre de références = # B3FF − # 400.

Il est évident que l'intervalle doit englober le signe et le nombre de décimales souhaités.

En réalité, il est plus pratique d'adapter un petit programme en fonction du cas traité, ainsi les méthodes les plus simples seront les plus rapides. Et n'oubliez pas que le même principe peut être appliqué aux chaînes de caractères.

```
              Exemple no 1

48  43  73  66  60  32  50  35  90  82  27  84  61  65  50  49  82  86  42  77
0  63  40  53  27  75  78  33  71  57  52  65  59  63  68  94  92  71  27  69
97  49  48  10  56  49  90  5  84  68  83  23  14  54  94  60  67  25  73  38
96  50  36  33  64  89  98  36  32  15  30  48  60  42  84  22  37  87  15  38
63  78  69  74  50  17  56  0  15  4  9  86  89  59  5  99  19  93  76  82  86

0  0  4  5  5  9  10  14  15  15  15  17  19  22  23  25  27  27  27  30  32
32  33  33  35  36  36  37  38  38  40  42  42  43  48  48  48  49  49  49  50
50  50  50  52  53  54  56  56  57  59  59  60  60  60  61  63  63  63  64  65
65  66  67  68  68  69  69  71  71  73  73  74  75  76  77  78  78  82  82  82
83  84  84  84  86  86  86  87  89  89  90  90  92  93  94  94  96  97  98  99
```

```
              Exemple no 2

TYPE    NOMBRE        NOMBRE    TYPE

0       28            1         9
1       50            14        8
2       28            19        6
3       61            28        0
4       57            28        2
5       70            32        7
6       19            50        1
7       32            57        4
8       14            61        3
9       1             70        5
10      87            87        10
```

```
              Exemple no 3

9.06915464  3.33571472  1.43994805  7.29736794  1.06552989  4.55143547
7.81271318  8.73512063  .833568346  1.63117823  9.47436209  4.13044493
4.36496412  7.58770384  9.86485931  3.93819672  1.31927404  7.6294218
.32408687  .27298971  5.59958257  1.87517452  .897003132  2.65929729
2.41475269  6.68306511  4.19080076  4.09887468  4.04681837  3.11372374
2.60807686  .277493441  8.9047549  9.00055685  2.7402673  .222209558
.347421469  2.67671513  6.29221008  3.24666553  5.44554587  9.62578462
5.09703856  2.17912061  7.73443318  7.93750498  8.74535773  2.98925437
3.89616659  3.60350284  8.46248359  6.53895772  3.24453528  9.89263709
1.34067816  6.96941779  4.08850729  4.07119905  6.03887533  9.41134206
4.71082191  4.91342883  1.4112888  6.00221205  3.14848414  3.80027884
7.64739843  4.59462794  1.70200566  1.2818223  .151647761  3.63001507
4.86721665  2.24428278  5.53617992  2.60569917  6.95044621  6.63976549
4.59823188  7.91804651  .919080115  8.99529191  7.58028903  5.28929683
4.28732499  2.43869281  7.76649576  6.78833279  8.79300056  8.17923882
7.0024613  8.04155171  6.67059337  6.82938871  .101142506  8.15129931
8.42799043  3.35047161  2.06768152  4.29380392

.11  .16  .23  .28  .28  .33  .35  .84  .9  .92  1.07  1.29  1.32  1.35  1.42
1.44  1.64  1.71  1.88  2.07  2.18  2.25  2.42  2.44  2.61  2.61  2.66  2.68
2.75  2.99  3.12  3.15  3.25  3.25  3.34  3.36  3.61  3.64  3.81  3.9  3.94
4.05  4.08  4.09  4.1  4.14  4.2  4.29  4.3  4.37  4.56  4.6  4.6  4.72  4.87
4.92  5.1  5.29  5.45  5.54  5.6  6.01  6.04  6.3  6.54  6.64  6.68  6.69
6.79  6.83  6.96  6.97  7.01  7.3  7.59  7.59  7.63  7.65  7.74  7.77  7.82
7.92  7.94  8.05  8.16  8.18  8.43  8.47  8.74  8.75  8.8  8.91  9  9.01  9.07
9.42  9.48  9.63  9.87  9.9
```

```
              Exemple no 4

2.94268388 -4.38213224  3.33725829 -4.35497223  4.68521435 -1.24717613
1.83316401 -3.80697683  2.91047321 -3.1255796  .910671915 -2.09101901
4.738545 -.210719606  1.98160826 -4.52500897  3.98530085 -1.41564654
.181417406 -2.97782237  3.08423753 -.715406499  2.64624828 -3.61077182
1.4043577 -.0908502326  1.49986113 -4.48243852  1.41609539 -3.42487598
1.17827117 -2.96612909  1.21220942 -2.04662761  .261617774 -2.76861238
3.56921428 -1.66790053  2.20447121 -1.16874207  2.5094523 -.17486057
1.05191105 -4.69808239  2.72113135 -2.17384898  4.64100111 -3.05180718
.103073974 -1.76083414

-4.69 -4.52 -4.48 -4.38 -4.35 -3.8 -3.61 -3.42 -3.12 -3.05 -2.97 -2.96 -2.76
-2.17 -2.09 -2.04 -1.76 -1.66 -1.41 -1.24 -1.16 -.71 -.21 -.17 -.09  .11  .19
.27  .92  1.06  1.18  1.22  1.41  1.42  1.5  1.84  1.99  2.21  2.51  2.65
2.73  2.92  2.95  3.09  3.34  3.57  3.99  4.65  4.69  4.74
```



UTILITEL

OU COMMENT FAIRE DES ECONOMIES AVEC SON MINITEL

Frédéric TARAUD

Utilisé conjointement avec la carte série décrite dans le numéro 14 de THEORIC, ce programme vous permettra de mettre en mémoire, de visualiser, puis de sauver les pages Vidéotext de votre serveur favori. Il vous sera en outre possible de les envoyer ensuite à un correspondant !

Vous pourrez à présent rester le temps qu'il vous plaira devant le petit écran de votre Minitel à relire des pages enregistrées auparavant à la ''va-vite'' sans qu'il ne vous en coûte un centime en communication, ni que votre ligne téléphonique ne soit encombrée.
Ce programme est compatible ORIC-1/ORIC ATMOS : une modification est à effectuer dans le programme Basic de chargement en ligne 130, pour les possesseurs d'ORIC-1.

MISE EN ŒUVRE

Tapez le programme UTILITEL, puis sauvez-le. Chargez le langage machine à l'aide du chargeur DATAS. Vérifiez les checksums et sauvegardez le machine à la suite du Basic. Connectez la carte série, et l'ORIC n'attend plus qu'un RUN pour vous présenter le menu :

ENREGISTREMENT
VISUALISATION
SUPPRESSION
SAUVEGARDE
CHARGEMENT
ENVOI DE PAGES
RAZ PAGES
FIN

Il vous suffit de taper n'importe quelle touche pour faire passer alternativement chacune de ces propositions en vidéo inverse. Un appui sur RETURN validera votre choix.

ENREGISTREMENT

Il vous faudra effectivement commencer par là. Si toutefois vous choisissez VISUALISATION, SUPPRESSION, ENVOI ou SAUVEGARDE sans pages en mémoire, le message ''Pas de pages en mémoire... '' vous ramenerait sur le bon chemin, c'est-à-dire au menu.
Quand cette option est validée, elle démarre automatiquement et l'inscription ''Enregistrement page 1'' apparaît. A partir de ce moment, tout octet pénétrant dans la prise téléphonique s'inscrit en mémoire.

Si une interruption de plus d'une seconde a lieu dans la transmission, le programme considérera que la page est terminée et rendra la main. Cette temporisation devrait suffire dans la plupart des cas, nous verrons quand même plus loin comment la modifier si nécessaire.
Notons qu'il est possible de définir la fin de l'espace qu'occuperont les pages en mémoire, et que le message ''plus de place... '' vous précisera si cette limite est atteinte.
La touche REPETITION pourra vous être utile (si le serveur l'utilise), car son action est de renvoyer la page actuellement à l'écran. Vous pourrez ainsi l'enregistrer APRES en avoir eu connaissance.
De toute façon, si vous faisiez l'acquisition d'une page indésirable, il vous suffirait de valider l'option.

SUPPRESSION

Après avoir indiqué le numéro de la page à supprimer, un simple appui sur RETURN ''rayera de la carte'' la page en question.

VISUALISATION

De la même façon que pour la suppression, vous rentrerez le numéro de la page. La page s'inscrira sur le Minitel avec en ligne 0 la mention ''Page XX'', XX correspondant au numéro choisi auparavant.

SAUVEGARDE

Ceci sauvegardera la zone mémoire des pages actuellement enregistrées en précisant à l'écran les adresses limites et en vous demandant le nom du fichier.

CHARGEMENT

Vous permettra de recharger un fichier sauvé par l'option précédente.
Remarquons que les deux dernières options pourront très facilement, et avec bonheur, être adaptées aux disquettes.

ENVOI DE PAGES

Vous pourrez, grâce à cette option, renvoyer des pages enregistrées à un de vos amis. Deux possibilités vous sont offertes :
— envoyer la totalité des pages en mémoire,
— envoyer certaines des pages en mémoire.
Dans le premier cas, votre correspondant pourra ''feuilleter'' vos pages en appuyant sur la barre d'espace de son Minitel, faire réapparaître la même page avec REPETITION, ou se déconnecter avec CNX/FIN.
Dans le second cas, c'est vous qui choisirez une à une les pages à transmettre et c'est également vous qui déconnecterez.

RAZ PAGES ET FIN

Ces deux commandes se passent évidemment de commentaires. Notez simplement que la RAZ ne fait que placer un double zéro en début de zone de stockage, et ne détruit donc pas les pages, contrairement à la commande SUPPRESSION (c'est toujours bon à savoir...).

QUELQUES DETAILS

Les pages sont stockées à partir de l'adresse # 2200. L'octet de poids fort, définissant la fin de zone de stockage, doit être placé en # 05. Ceci est effectué en ligne 200. Dans le cas présent, cette zone de stockage s'étendra de # 2200 à # 9800, ce qui nous laisse environ 30 ko, soit 20 à 30 pages...
Le nombre maximum de pages a été ici fixé arbitrairement à 30 par l'instruction E=31:DIM T(E) en ligne 140.
Signalons que le programme vous signalera si la capacité de ce tableau est dépassée, et stoppera tout nouvel enregistrement. Vous pourrez toujours augmenter cette capacité, mais attention, le HIMEM est en # 2000, et un tableau est gourmand en mémoire.

PRINCIPE DU STOCKAGE

La méthode utilisée ici ne brille certainement pas par son originalité, ce qui prouve au moins qu'elle est efficace...
Chaque page est précédée par 4 octets. Les deux premiers contiennent l'adresse de la page suivante, et les deux autres la longueur de la page. Un double zéro signale la fin de la dernière page. Cette adresse est contenue dans la variable EOF, alors que les adresses des différentes pages sont contenues dans le tableau T. Ainsi, T(3) sera égal à l'adresse de la troisième page, ou plus exactement, il faudra ajouter 4 pour obtenir l'adresse du premier octet de cette page.
Le tableau T ainsi que EOF sont mis à jour par un sous-programme Basic situé de 1170 à 1290.

LES ROUTINES MACHINE

La première est la routine d'enregistrement et se situe de # 2000 à # 202D. Son mécanisme est le suivant : après avoir été lancée par le Basic (ligne 620), elle attend l'arrivée d'un octet. Si rien n'arrive au bout d'une seconde, elle revient au Basic. Si celui-ci ne détecte aucun octet enregistré, il relance la routine et ainsi de suite tant que rien ne se passe. Par contre, si une page est envoyée, la routine ne reviendra au Basic que lorsque le dernier octet sera transmis, c'est-à-dire après une seconde de "silence". Le Basic, voyant qu'au moins un octet a été reçu, considérera la transmission terminée.
Comme il a été dit plus haut, il vous sera possible de modifier ceci de telle sorte que l'enregistrement ne cesse que lorsqu'un nombre minimum d'octets est enregistré. Pour ce faire, modifiez la ligne 650 en conséquence : IF DEEK(0)<D+X avec X nombre d'octets à enregistrer avant fin de page −4 (ici X=14). Ceci peut être utile avec des serveurs qui envoient d'abord un effacement de page (# 0C) puis, après seulement quelques secondes, la page.
Avant appel, 00 et 01 doivent contenir l'adresse de stockage, 05 le poids fort de fin de mémoire et 04 zéro. Si la fin de mémoire est détectée, 04 passera à 1, et un retour immédiat au Basic s'effectuera. Les octets 02 et 03 sont utilisés pour la temporisation d'une seconde.
La seconde routine permet des transferts de blocs. Elle est utilisé par l'option SUPPRESSION. Pour son utilisation, différents pointeurs devront être positionnés :

06 et 07 : adresse de départ.
08 et 09 : adresse de destination.
0A et 0B : longueur du bloc.

Les connaisseurs reconnaîtront la routine de transfert de l'ORIC, recopiée ici pour éviter des problèmes de compatibilité.
La troisième routine permet, quant à elle, l'envoi sur le Minitel d'une zone mémoire avec les pointeurs suivants :

06 et 07 : adresse de départ.
0A et 0B : longueur.

Elle se trouve de # 204A à # 2066.
Ces trois routines utilisent une sous-routine d'envoi d'un octet sur le Minitel située de # 2067 à # 207B.
Les dernières routines (de # 207C à # 2121) concernent la gestion de l'envoi des pages : envoi de la porteuse, affichage de messages, test de clavier, etc...
La zone de # 2122 à # 21D4 contient les différents messages à afficher et les séquences à transmettre.
Ces explications devraient vous permettre de saisir le fonctionnement de ce programme, et vous aider à le développer afin de le rendre encore plus performant.

```
10 '*********************
20 '*                   *
30 '*     UTILITEL      *
40 '*                   *
50 '* Frederic TARAUD   *
60 '*                   *
70 '*  le 04/03/86      *
80 '*                   *
90 '*********************
100 '
110 '  Initialisations
120 '
130 CLS:PAPER0:INK7:HIMEM#2000
140 E=31:DIM T(E):POKE#26A,14
150 IF DEEK(#2017)=#380 THEN 170
160 CLOAD"Utilitel"
170 POKE#381,#01 '
180 POKE#382,#6B ' Init UART
190 POKE#383,#38 '
200 POKE#05 ,#98 ' OPS Fin Memoire
```



```
210 POKE#04 ,#00 ' Flag fin Memoire
220 DOKE#2200,00 ' Debut de
230 DOKE#2202,00 ' Fichier
240 '
250 '        Menu
260 CLS
270 PLOT10,05,"Enregistrement"
280 PLOT10,06,"Visualisation "
290 PLOT10,07," Suppression  "
300 PLOT10,08," Chargement   "
310 PLOT10,09," Sauvegarde   "
320 PLOT10,10,"Envoi de pages"
330 PLOT10,11,"   RAZ pages  "
340 PLOT10,12,"     Fin      "
350 I=5
360 PLOT8,I,0:PLOT9,I,23:PLOT25,I,16
370 PLOT8,N,8:PLOT9,N,8
380 GETZ$:Z=ASC(Z$):IF Z=13 THEN 420
390 I=I+1:N=I-1
400 IF I=13 THEN 350
410 GOTO 360
420 A=I-4:IF A=8 THEN POKE#26A,15:END
430 IF A=7 THEN 460
440 ON A GOSUB 530,730,970,1430,1310,1530
450 GOTO 260
460 '
470 '         RAZ pages
480 '
490 CLS:PRINT"  RAZ Pages":PRINT
500 PRINT"Confirmation...(O/N) :"
510 GETZ$:IF Z$="O" THEN PRINT"RAZ...":WAIT50:GOTO170
520 GOTO 260
530 '
540 '  Enregistrement pages
550 '
560 CLS
570 IF PEEK(4)=1 THEN RETURN
580 IF DEEK(#2200)=0 THEN D=#2200:N=0:GOTO 610
590 DRAP=0:GOSUB 1240:IF DRAP=1 THEN WAIT 100:RETURN
600 D=EOF
610 DOKE 0,D+4
620 CLS:PRINTCHR$(27)CHR$(140)"Enregistrement page ";N+1
630 CALL#2000
640 IF PEEK(4)=1 THEN PRINT"Plus de place...":GOTO 660
650 IF DEEK(0)<D+14 THEN 630
660 F=DEEK(0)
670 L=F-(D+4)
680 DOKED  ,F'Adr.debut page suivante
690 DOKED+2,L'Longueur page
700 DOKEF  ,0'Marque fin de fichier
710 PRINT:PRINT"Longueur enregistrement :";L;"Octets":PRINT
720 PRINT"Autre enregistrement (O/N)";:GETZ$:IFZ$="O"THEN 560 ELSE RETURN
730 '
740 '    Envoi des pages
750 '
760 ENV=0:CLS
770 FLAG=0:GOSUB 1170:IF FLAG=1 THEN WAIT 50:RETURN
780 CLS:PRINT"  Envoi de pages":PRINT
790 PRINT"Page 1 a ";N;:INPUT P
800 IF P<1 OR P>N THEN 790
810 DIZ=INT(P/10)
820 UNIT=P-(10*DIZ)
830 POKE#21C4,DIZ +#30
840 POKE#21C5,UNIT+#30
850 CALL#2112 'Blanck
```

```
860 CALL#211A 'Message No de page
870 D=T(P)+4
880 L=DEEK(T(P)+2):GOSUB 920
890 IF ENV=1 THEN RETURN
900 PRINT:PRINT"Autre pages (O/N)";:GETZ$:IFZ$="O" THEN 780 ELSE RETURN
910 ' SP Envoi
920 POKE#380,#0C 'Efface le Minitel
930 DOKE #06,D    'Depart
940 DOKE #0A,L    'Longueur
950 CALL #204A    'Envoi sur Minitel
960 RETURN
970 '
980 '       Suppression
990 '
1000 CLS
1010 PRINT"   Suppression":PRINT
1020 FLAG=0:GOSUB 1170:IF FLAG=1 THEN WAIT 50:RETURN
1030 PRINT"Page 1 a ";N;:INPUT P
1040 IF P<1 OR P>N THEN  1030
1050 IF P=N THEN DOKE T(P),0:GOTO1160
1060 L=(EOF-T(P+1))+2
1070 :DOKE#06,T(P+1) 'Depart
1080 :DOKE#08,T(P)   'Destination
1090 :DOKE#0A,L      'Longueur
1100 :CALL#202E      'Transfert
1110 FOR X=P TO N-1
1120 L=DEEK(T(X)+2)  'Reajustement
1130 T(X+1)=T(X)+L+4 'des descripteurs
1140 DOKE T(X),T(X+1)'
1150 NEXT X
1160 PRINT:PRINT"Autre suppression (O/N)";:GETZ$:IFZ$="O"THEN1000ELSERETURN
1170 '
1180 '        Sous programme :
1190 ' Nombre de pages enregistrees
1200 '  Determination des adresses
1210 '  et controle presence pages
1220 '
1230 IF DEEK(#2200)=0 THEN FLAG=1:PRINT"Pas de pages enregistrees...":RETURN
1240 T(1)=#2200:N=2
1250 T(N)=DEEK(T(N-1))
1260 IF DEEK(T(N))=0 THEN EOF=T(N):N=N-1:GOTO 1290
1270 IF N+1=E AND DRAP=0 THEN PRINT"Capacite depasee...":DRAP=1:RETURN
1280 N=N+1:GOTO 1250
1290 RETURN
1300 '
1310 '         Sauvegarde
1320 '
1330 CLS
1340 PRINT:PRINT" Sauvegarde":PRINT
1350 FLAG=0:GOSUB 1170:IF FLAG=1 THEN WAIT 50:RETURN
1360 PRINT CHR$(27)CHR$(140)  N"Pages"
1370 PRINT:PRINT"De #2200 a ";HEX$(EOF+1):PRINT
1380 INPUT"Nom :";N$
1390 PRINT:PRINT"Appuyer sur une touche":GETZ$
1400 CSAVE N$,A#2200,E (EOF+1)
1410 POKE#381,0:POKE#382,#6B:POKE#383,#38
1420 RETURN
1430 '
1440 '         Chargement
1450 '
1460 CLS
1470 PRINT"         Chargement":PRINT
1480 PRINT"Nom :";:INPUT N$:PRINT
1490 PRINT"  Appuyer sur une touche":GETZ$
1500 CLOAD N$
```

```
1510 POKE#381,0:POKE#382,#6B:POKE#383,#38
1520 RETURN
1530 '
1540 '    Envoi vers correspondant
1550 '
1560 CLS
1570 FLAG=0:GOSUB 1170:IF FLAG=1 THEN WAIT 50:RETURN
1580 DEF USR=#20B9
1590 CLS:ENV=1
1600 PRINT"     Envoi vers un correspondant"
1610 PRINT:PRINT"       Envoi de toutes les pages"
1620 PRINT"           en memoire (O/N) ?"
1630 PRINT:PRINT"          (ESC pour le menu)"
1640 REPEAT:GET Z$:Z=ASC(Z$):UNTIL Z=79 OR Z=78 OR Z=27
1650 IF Z=78 THEN GOTO 1830
1660 IF Z=79 THEN GOTO 1690
1670 RETURN
1680 '
1690 'Envoi toutes les pages
1700 '
1710 GOSUB 1950
1720 P=1
1730 PRINT@10,11;"Transmission page ";P
1740 GOSUB 810'Envoi page
1750 A=USR(Z) 'Attente reponse :
1760 IF A=1 THEN P=P-1 'Repetition
1770 IF A=-1 THEN 1800 'CNX/FIN
1780 P=P+1               'ESPACE
1790 IF P<>N+1 THEN 1730
1800 GOSUB 2020
1810 GOTO 1590
1820 '
1830 'Envoi une page a la fois
1840 '
1850 GOSUB 1950
1860 PRINT
1870 PRINT"Page 1 a ";N;:INPUT P
1880 IF P<1 OR P>N THEN 1870
1890 GOSUB 810 'Envoi
1900 PRINT"Autre page (O/N)"
1910 REPEAT:GET Z$:UNTIL Z$="O" OR Z$="N"
1920 IF Z$="O" THEN 1870
1930 GOSUB 2020
1940 GOTO 1590
1950 '
1960 ' SP Envoi Porteuse
1970 '
1980 PRINT:PRINT"Envoi porteuse..."
1990 CALL#207C
2000 PRINT:PRINT"Correspondant pret"
2010 RETURN
2020 '
2030 ' SP deconnection
2040 '
2050 PRINT:PRINT"Fin de transmission..."
2060 CALL #20AA
2070 PRINT"Deconnection...":WAIT 100
2080 RETURN
```

```
10 '
20 '  Chargement DATAS
30 '
40 '     UTILITEL
```

```
50 '
60 CLS:CHKS=0:C=1:LIGNE=200
70 PRINT"Voulez-vous l'imprimante ?"
80 GET Z$:IF Z$="O" THEN POKE#2F1,128 ELSE POKE#2F1,0
90 FOR I=#2000 TO #21DF
100 READ D$:D=VAL("#"+D$):CHKS=CHKS+D*C:C=C+1:POKE I,D
110 IFC=17THEN PRINT"Ligne ";LIGNE;" CHKS=";CHKS:LIGNE=LIGNE+10:CHKS=0:C=1
120 NEXT
130 ' ORIC-1 :DOKE#20E3,#DF15
140 PRINT"<S> pour sauver"
150 PRINT"Autre touche pour arreter"
160 GET Z$
170 IF Z$<>"O" THEN END
180 CSAVE"Utilitel",A#2000,E#21DF
190 END
200 DATA A9,00,85,02,85,03,AD,81,03,29,08,D0,09,E6,02,D0
210 DATA F5,E6,03,D0,F1,60,AD,80,03,A0,00,91,00,E6,00,D0
220 DATA DF,E6,01,A5,01,C5,05,D0,D7,A9,01,85,04,60,A2,00
230 DATA A0,00,C4,0A,D0,04,E4,0B,F0,0F,B1,06,91,08,C8,D0
240 DATA F1,E6,07,E6,09,E8,4C,32,20,60,A2,00,A0,00,C4,0A
250 DATA D0,04,E4,0B,F0,10,B1,06,85,16,20,67,20,C8,D0,EE
260 DATA E6,07,E8,4C,4E,20,60,8A,48,98,48,AD,81,03,29,10
270 DATA F0,F9,A5,1E,8D,80,03,68,A8,68,AA,60,A9,22,A0,21
280 DATA 20,E9,20,20,FA,20,C9,13,D0,F9,20,FA,20,C9,53,D0
290 DATA F2,A9,29,A0,21,20,E9,20,A9,35,A0,21,20,E9,20,20
300 DATA FA,20,C9,20,D0,F9,20,12,21,60,A9,8E,A0,21,20,E9
310 DATA 20,A9,D1,A0,21,20,E9,20,60,A9,62,A0,21,20,E9,20
320 DATA 20,FA,20,C9,13,D0,0F,20,FA,20,C9,43,D0,02,F0,10
330 DATA C9,49,D0,EC,F0,07,C9,20,D0,E6,A9,00,2C,A9,FF,2C
```

```
340 DATA A9,01,20,24,DF,20,12,21,60,85,02,84,03,A0,00,B1
350 DATA 02,F0,06,20,05,21,C8,D0,F6,60,AD,81,03,29,08,F0
360 DATA F9,AD,80,03,60,48,AD,81,03,29,10,F0,F9,68,8D,80
370 DATA 03,60,A9,C8,A0,21,20,E9,20,60,A9,BB,A0,21,20,E9
380 DATA 20,60,1B,39,6F,1B,39,68,00,1B,3B,61,58,53,1B,3B
390 DATA 61,5B,51,0C,00,1F,40,41,1B,51,1B,48,20,20,20,20
400 DATA 20,20,20,20,20,20,20,20,20,20,54,61,70,65,7A,20
410 DATA 45,53,50,41,43,45,20,20,20,20,20,20,20,20,20,20
420 DATA 20,00,1F,40,41,1B,51,20,2A,45,53,50,3A,53,75,69
430 DATA 74,65,2A,43,4E,58,2F,46,49,4E,3A,53,74,6F,70,2A
440 DATA 52,45,50,45,54,3A,52,65,76,6F,69,72,2A,00,1F,40
450 DATA 41,1B,51,20,20,20,20,20,20,20,46,69,6E,20,64,65
460 DATA 20,74,72,61,6E,73,6D,69,73,73,69,6F,6E,2E,2E,2E
470 DATA 20,20,20,20,20,20,20,20,20,0A,00,1F,40,55,20,50
480 DATA 61,67,65,20,30,32,0A,00,1F,40,41,1B,51,20,12,67
490 DATA 00,1B,39,67,00,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA
```



Checksums :

```
Ligne  200  CHKS= 13218
Ligne  210  CHKS= 15477
Ligne  220  CHKS= 13290
Ligne  230  CHKS= 16190
Ligne  240  CHKS= 12221
Ligne  250  CHKS= 16676
Ligne  260  CHKS= 11228
Ligne  270  CHKS= 14822
Ligne  280  CHKS= 19240
Ligne  290  CHKS= 12464
Ligne  300  CHKS= 15517
Ligne  310  CHKS= 14317
Ligne  320  CHKS= 15309
Ligne  330  CHKS= 18318
Ligne  340  CHKS= 11019
Ligne  350  CHKS= 15113
Ligne  360  CHKS= 16495
Ligne  370  CHKS= 16689
Ligne  380  CHKS= 8219
Ligne  390  CHKS= 5794
Ligne  400  CHKS= 9060
Ligne  410  CHKS= 5164
Ligne  420  CHKS= 9983
Ligne  430  CHKS= 11004
Ligne  440  CHKS= 9616
Ligne  450  CHKS= 8954
Ligne  460  CHKS= 12009
Ligne  470  CHKS= 5694
Ligne  480  CHKS= 6721
Ligne  490  CHKS= 28951
```


voir page **8**

```
2000-   A9 00       LDA #$00
2002-   85 02       STA $02
2004-   85 03       STA $03
2006-   AD 81 03    LDA $0381
2009-   29 08       AND #$08
200B-   D0 09       BNE $2016
200D-   E6 02       INC $02
200F-   D0 F5       BNE $2006
2011-   E6 03       INC $03
2013-   D0 F1       BNE $2006
2015-   60          RTS
2016-   AD 80 03    LDA $0380
2019-   A0 00       LDY #$00
201B-   91 00       STA ($00),Y
201D-   E6 00       INC $00
201F-   D0 DF       BNE $2000
2021-   E6 01       INC $01
2023-   A5 01       LDA $01
2025-   C5 05       CMP $05
2027-   D0 D7       BNE $2000
2029-   A9 01       LDA #$01
202B-   85 04       STA $04
202D-   60          RTS
```

Enregistrement et tempo 1 seconde

04 est mis à 1 si capacité mémoire insuffisante

```
202E-   A2 00       LDX #$00
2030-   A0 00       LDY #$00
2032-   C4 0A       CPY $0A
2034-   D0 04       BNE $203A
2036-   E4 0B       CPX $0B
2038-   F0 0F       BEQ $2049
203A-   B1 06       LDA ($06),Y
203C-   91 08       STA ($08),Y
203E-   C8          INY
203F-   D0 F1       BNE $2032
2041-   EE 07       INC $07
2043-   E6 09       INC $09
2045-   E8          INX
2046-   4C 32 20    JMP $2032
2049-   60          RTS
```

Transfert de blocs
06,07 : Départ
08,09 : Arrivée
0A,0B : Longueur

```
204A-   A2 00       LDX #$00
204C-   A0 00       LDY #$00
204E-   C4 0A       CPY $0A
2050-   D0 04       BNE $2056
2052-   E4 0B       CPX $0B
2054-   F0 10       BEQ $2066
2056-   B1 06       LDA ($06),Y
2058-   85 16       STA $16
205A-   20 67 20    JSR $2067
205D-   C8          INY
205E-   D0 EE       BNE $204E
2060-   E6 07       INC $07
2062-   E8          INX
2063-   4C 4E 20    JMP $204E
2066-   60          RTS
```

Envoi sur le Minitel d'une zone mémoire
06,07 : Départ
0A,0B : Longueur

```
2067-   8A          TXA
2068-   48          PHA
2069-   98          TYA
206A-   48          PHA
206B-   AD 81 03    LDA $0381
206E-   29 10       AND #$10
2070-   F0 F9       BEQ $206B
2072-   A5 16       LDA $16
2074-   8D 80 03    STA $0380
2077-   68          PLA
2078-   A8          TAY
2079-   68          PLA
207A-   AA          TAX
207B-   60          RTS
```

Sous-routine d'envoi d'un octet avec sauvegarde du contexte

```
207C-   A9 22       LDA #$22
207E-   A0 21       LDY #$21
2080-   20 E9 20    JSR $20E9
```

Retournement du Modem et envoi porteuse

```
2083-   20 FA 20    JSR $20FA
2086-   C9 13       CMP #$13
2088-   D0 F9       BNE $2083
208A-   20 FA 20    JSR $20FA
208D-   C9 53       CMP #$53
208F-   D0 F2       BNE $2083
```

Attente CNX/FIN

```
2091-   A9 29       LDA #$29
2093-   A0 21       LDY #$21
2095-   20 E9 20    JSR $20E9
```

Aiguillages
Prise→Ecran
Clavier→Prise

```
2098-   A9 35       LDA #$35
209A-   A0 21       LDY #$21
209C-   20 E9 20    JSR $20E9
```

Affichage message

```
209F-   20 FA 20    JSR $20FA
20A2-   C9 20       CMP #$20
20A4-   D0 F9       BNE $209F
```

Attente barre espace Minitel

```
20A6-   20 12 21    JSR $2112
20A9-   60          RTS
```

Efface ligne 0 Minitel
Retour Basic

```
20AA-   A9 8E       LDA #$8E
20AC-   A0 21       LDY #$21
20AE-   20 E9 20    JSR $20E9
20B1-   A9 D1       LDA #$D1
20B3-   A0 21       LDY #$21
20B5-   20 E9 20    JSR $20E9
20B8-   60          RTS
```

Envoi message : "Fin de transmission" et déconnexion des deux Minitels.

Retour Basic

```
20B9-   A9 62       LDA #$62
20BB-   A0 21       LDY #$21
20BD-   20 E9 20    JSR $20E9
20C0-   20 FA 20    JSR $20FA
20C3-   C9 13       CMP #$13
20C5-   D0 0F       BNE $20D6
20C7-   20 FA 20    JSR $20FA
20CA-   C9 43       CMP #$43
20CC-   D0 02       BNE $20D0
20CE-   F0 10       BEQ $20E0
20D0-   C9 49       CMP #$49
20D2-   D0 EC       BNE $20C0
20D4-   F0 07       BEQ $20DD
20D6-   C9 20       CMP #$20
20D8-   D0 E6       BNE $20C0
20DA-   A9 00       LDA #$00
20DC-   2C A9 FF    BIT $FFA9
20DF-   2C A9 01    BIT $01A9
20E2-   20 24 DF    JSR $DF24
20E5-   20 12 21    JSR $2112
20E8-   60          RTS
```

USR
Affiche message puis attend Espace, CNX/FIN ou SUITE
*** Si espace, renvoi Ø dans la variable**
*** Si CNX/FIN, renvoi −1 dans la variable**
*** Si suite, renvoi +1 dans la variable**
Sort par un JSR$IF24 (ATMOS) ou par JSR$DF15 (ORIC-1)
(Rond flottant le registre A)

SOUS PROGRAMMES :

```
20E9-   85 02       STA $02
20EB-   84 03       STY $03
20ED-   A0 00       LDY #$00
20EF-   B1 02       LDA ($02),Y
20F1-   F0 06       BEQ $20F9
20F3-   20 05 21    JSR $2105
20F6-   C8          INY
20F7-   D0 F6       BNE $20EF
20F9-   60          RTS
```

Envoi un message ou une séquence dont l'adresse est dans A(−) et Y(+).
La séquence doit se terminer par un 00.

```
20FA-   AD 81 03    LDA $0381
20FD-   29 08       AND #$08
20FF-   F0 F9       BEQ $20FA
2101-   AD 80 03    LDA $0380
2104-   60          RTS
```

Sous-routine : reçoit un octet dans A.

```
2105-   48          PHA
2106-   AD 81 03    LDA $0381
2109-   29 10       AND #$10
210B-   F0 F9       BEQ $2106
210D-   68          PLA
210E-   8D 80 03    STA $0380
2111-   60          RTS
```

Sous-routine : envoi un octet mis dans A

```
2112-   A9 C8       LDA #$C8
2114-   A0 21       LDY #$21
2116-   20 E9 20    JSR $20E9
2119-   60          RTS
```

Efface ligne 0 Minitel.

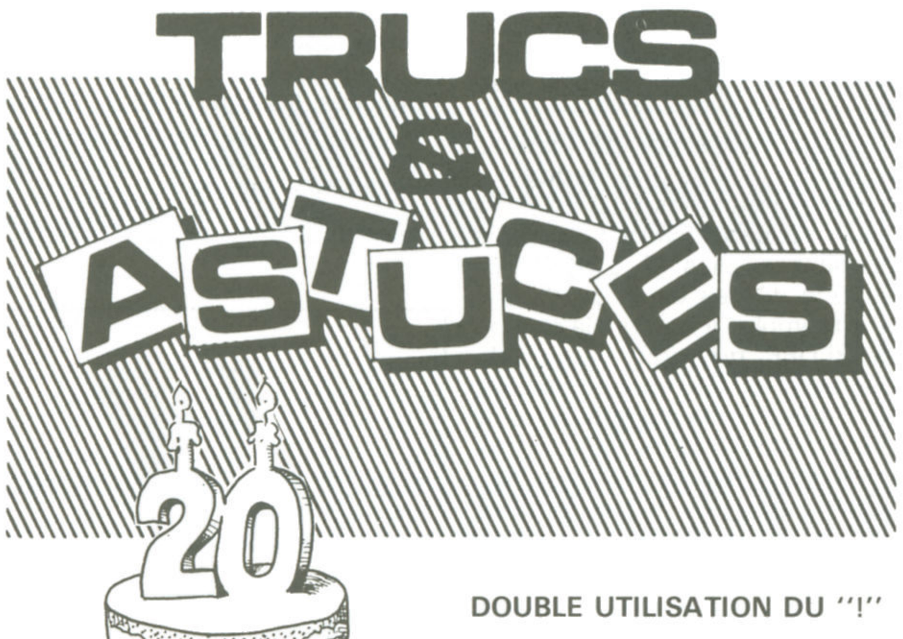
```
211A-   A9 BB       LDA #$BB
211C-   A0 21       LDY #$21
211E-   20 E9 20    JSR $20E9
2121-   60          RTS
```

Affiche "Page xx"

```
122 1B 39 6F 1B 39 68 00 1B ¯9o¯9h¯¯
212A 3B 61 58 53 1B 3B 61 5B ;aXS¯;a↑
```

```
2132 51 0C 00 1F 40 41 1B 51 Q¯¯¯@A¯Q
213A 1B 48 20 20 20 20 20 20 ¯H
2142 20 20 20 20 20 20 20 20
214A 54 61 70 65 7A 20 45 53 Tapez ES
2152 50 41 43 45 20 20 20 20 PACE
215A 20 20 20 20 20 20 20 00        ¯
2162 1F 40 41 1B 51 20 2A 45 ¯@A¯Q *E
216A 53 50 3A 53 75 69 74 65 SP:Suite
2172 2A 43 4E 58 2F 46 49 4E *CNX/FIN
217A 3A 53 74 6F 70 2A 52 45 :Stop*RE
2182 50 45 54 3A 52 65 76 6F PET:Revo
218A 69 72 2A 00 1F 40 41 1B ir*¯¯@A¯
2192 51 20 20 20 20 20 20 20 Q
219A 4E 69 6E 20 64 65 20 74 Fin de t
21A2 72 61 6E 73 6D 69 73 73 ransmiss
21AA 69 6F 6E 2E 2E 2E 20 20 ion...
21B2 20 20 20 20 20 20 20 0A        ¯
21BA 00 1F 40 55 20 50 61 67 ¯¯@U Pag
21C2 65 20 30 32 0A 00 1F 40 e 02¯¯¯@
21CA 41 1B 51 20 12 67 00 1B A¯Q ¯g¯¯
21D2 39 67 00 EA EA EA EA EA 9g¯¯¯¯¯¯
```

Séquences et messages



THEORIC arrive au numéro 20 avec ses fidèles lecteurs et abonnés. A l'intention des plus "jeunes" (des moins expérimentés, aurions nous dû dire), nous avons récapitulé tout un tas de trucs et astuces qui vont paraître aux plus chevronnés bien connus mais... de l'indulgence, voyons ! Cette page est donc réservée aux débutants.

MEMOIRE DISPONIBLE

Pour connaître à tout instant, dans un programme, la quantité de mémoire disponible, il suffit d'écrire X = FRE(0). X contiendra alors le nombre d'octets encore utilisables. Un simple test sur cette variable évitera au programme de se planter par manque de mémoire.

DOUBLE UTILISATION DU "!"

Quand on possède un drive, le ! est utilisé pour les commandes du DOS. Si l'on veut faire appel à un sous-programme ou à une nouvelle fonction Basic utilisant également le !, il suffit d'écrire :

DOKE # 2F5, adresse fonction

pour "brancher" le ! sur la fonction, ou :

DOKE # 2F5, # 400 (Jasmin)
(# 4C4 Microdisc)

pour revenir au DOS.

PROGRAMME PROTEGE

Pour bloquer le lancement automatique d'un programme Basic protégé et pouvoir le lister (ou le transférer sur disquette), il suffit, avant de le charger à partir de la cassette, de faire :

POKE # 500, # FF puis CLOAD

Après le chargement, vous obtiendrez "SYNTAX ERROR". Il ne reste plus qu'à faire un POKE # 500,0 pour pouvoir lancer le programme par RUN.

SUPPRIMER CAPS

Le message CAPS, qui signale que l'ORIC est en mode "majuscules", peut être supprimé, sans forcer pour autant l'ORIC, en minuscules (bascule CTRL T). Il suffit de faire :

POKE 48035,0

La magie aidant, POKE 48035,7 fera revenir le CAPS. On a tout simplement joué avec l'attribut couleur encre...

TRUQUER LE RESET

Le poussoir de RESET situé sous l'ORIC permet de reprendre, en principe, le contrôle du fonctionnement de la machine. Pour interdire à l'utilisateur la lecture du listing, certains éditeurs adoptaient, au début de l'ORIC, l'astuce suivante :

DOKE # 1B, # F42D (ORIC-1)
DOKE # 1B, # F88F (ATMOS)

L'appui sur RESET effectue alors une remise à zéro complète de la machine qui se retrouve dans le même état que si on venait de la mettre sous tension.
Si l'on veut bloquer le RESET, sans pour autant vider la mémoire, on peut adopter l'astuce suivante :

POKE # 22B,96 (ORIC-1)
POKE # 247,96 (ATMOS)

Eric VIEL

LES PIEDS DANS LE PLAT

C'est ce mois-ci que se termine notre programme FREDDY LEAPPY. Eh oui, tout a une fin... Arrêtons là les considérations sentimentales et donnons plutôt le menu. Nous allons faire clignoter les points que doit manger Freddy, et nous allons orchestrer l'ensemble des routines que nous avons étudiées jusqu'à présent.

LA ROUTINE REDEF

C'est elle qui fait clignoter les super-points que doit manger Freddy. Il faut tout d'abord "filtrer" l'entrée dans cette routine, car on risque de voir clignoter les points à une vitesse trop importante, et cela ne représenterait plus rien. C'est l'intérêt des lignes 614 à 618, qui, en se servant d'une mémoire nommée RD, n'autorisent l'accès à cette routine qu'une fois sur trois. Les lignes 619 et 620 remettent à zéro RD, pour le prochain accès.
Le clignotement sera réalisé d'une manière fort simple. Au lieu de remplacer en mémoire tous les caractères "e" par d'autres caractères dont la redéfinition graphique serait différente, nous allons modifier directement la représentation graphique de ces caractères, dans la mémoire où elle est stockée. Le manuel de l'ORIC vous indique la façon de calculer cette adresse : 46080 + 8* le code ASCII du caractère à redéfinir. Appliquons cette formule à la lettre "e", code ASCII 101, et nous trouvons 46888. Il suffira donc de faire alterner trois motifs dans ces cases mémoires pour donner une impression de clignotement. Ce sont d'ailleurs les lignes 621-630 qui jouent ce rôle. Suivant le valeur de RD2, on oriente le programme vers les lignes 631 ou 643 ou 654, qui redéfinissent donc le caractère "e".

LA ROUTINE SUITE

C'est elle qui orchestre le programme. Elle appelle, tour à tour, toutes les sous-routines que nous avons créées au cours des quelques mois que nous avons passés ensemble.
Elle commence par déplacer le monstres, vérifie que Freddy n'a pas été tué (lignes 812-814) ou qu'il n'a pas dévoré toutes les pastilles (lignes 815-817). Si aucune de ces deux conditions n'est remplie, il exécute la routine de REDEF, que nous venons de commenter. Commence ensuite une boucle de ralentissement (lignes 819 à 828) qui, comme son nom l'indique, ralentit l'ensemble du jeu. Vous pouvez essayer de modifier la valeur $ 13 en ligne 827 afin d'accélérer ou de ralentir le jeu. Nous testons ensuite si la touche ESC a été pressée (pour le suicide...) et, dans le cas contraire, on repart au début...

Voilà, c'est terminé, je vous ai détaillé entièrement le programme FREDDY LEAPPY, de manière approfondie et j'espère que cela vous aura permis de comprendre comment programmer en Assembleur. Le progamme proposé là est évidemment susceptible d'être nettement amélioré, mais j'ai préféré expliquer un programme un peu simple de par sa conception afin qu'il puisse être abordable par un grand nombre de lecteurs, les "pro" me le pardonneront volontiers (j'ose espérer...).
Au revoir à tous, et n'hésitez pas à envoyer à THEORIC vos plus belles réalisations... en Assembleur.

NOTE
En annexe, le listing des déclarations diverses et une routine permettant l'initialisation de toutes les mémoires est donné. Il permettra à ceux qui ont tapé les listings chaque mois de posséder le listing source complet.
Les lignes 1 à 86 ne seront pas publiées. Il s'agit de REM ! Entrez donc 86 lignes avec pour seule instruction le point virgule. Elles sont obligatoires.

```
53A5              611 :
53A5              612 ;          REDEF
53A5              613 ;
53A5   EE 0B 4F   614 REDEF       INC RD
53A8   AD 0B 4F   615             LDA RD
53AB   C9 03      616             CMP #3
53AD   F0 01      617             BEQ >1
53AF   60         618             RTS
53B0   A9 00      619 >1          LDA #00
53B2   8D 0B 4F   620             STA RD
53B5   EE 0C 4F   621             INC RD2
53B8   AD 0C 4F   622             LDA RD2
53BB   C9 01      623             CMP #1
53BD   F0 2C      624             BEQ >1
53BF   C9 03      625             CMP #3
53C1   F0 45      626             BEQ >3
53C3   C9 02      627             CMP #2
53C5   F0 05      628             BEQ >2
53C7   A9 00      629             LDA #0
53C9   8D 0C 4F   630             STA RD2
53CC   A9 00      631 >2          LDA #00
53CE   8D 28 B7   632             STA 46888
53D1   8D 29 B7   633             STA 46889
53D4   8D 2F B7   634             STA 46895
53D7   8D 2E B7   635             STA 46894
53DA   A9 1E      636             LDA #30
53DC   8D 2A B7   637             STA 46890
53DF   8D 2D B7   638             STA 46893
53E2   A9 12      639             LDA #18
53E4   8D 2B B7   640             STA 46891
53E7   8D 2C B7   641             STA 46892
53EA   60         642             RTS
53EB   A9 3F      643 >1          LDA #63
53ED   8D 28 B7   644             STA 46888
```

```
53F0    8D 2F B7  645         STA 46895
53F3    A9 21     646         LDA #33
53F5    8D 29 B7  647         STA 46889
53F8    8D 2A B7  648         STA 46890
53FB    8D 2B B7  649         STA 46891
53FE    8D 2C B7  650         STA 46892
5401    8D 2D B7  651         STA 46893
5404    8D 2E B7  652         STA 46894
5407    60        653         RTS
5408    A9 00     654 >3      LDA #0
540A    8D 28 B7  655         STA 46888
540D    8D 29 B7  656         STA 46889
5410    8D 2A B7  657         STA 46890
5413    8D 2D B7  658         STA 46893
5416    8D 2E B7  659         STA 46894
5419    8D 2F B7  660         STA 46895
541C    A9 0C     661         LDA #12
541E    8D 2B B7  662         STA 46891
5421    8D 2C B7  663         STA 46892
5424    60        664         RTS
```

```
FIN DE LA PASSE 1

Fin des labels:$45C7

0000              87          LST
0000              88          ORG $4F50
4F50   A9 45      89 INI      LDA #$45                ; INITIAL.
4F52   85 00      90          STA $00
4F54   85 02      91          STA $02
4F56   A9 BD      92          LDA #$BD
4F58   85 01      93          STA $01
4F5A   85 03      94          STA $03
4F5C   A9 01      95          LDA #01
4F5E   8D 00 4F   96          STA CTRL
4F61   A9 00      97          LDA #00
4F63   8D 07 4F   98          STA D/L
4F66   A9 28      99          LDA #40
4F68   8D 06 4F   100         STA H/L
4F6B   A9 37      101         LDA #$37
4F6D   85 04      102         STA $04
4F6F   A9 BF      103         LDA #$BF
4F71   85 05      104         STA $05
4F73   A9 00      105         LDA #00
4F75   8D 08 4F   106         STA ETAGE
4F78   8D 0A 4F   107         STA PAIR
4F7B   8D 0B 4F   108         STA RD
4F7E   8D 0C 4F   109         STA RD2
4F81   A9 3C      110         LDA #60
4F83   85 06      111         STA 06
4F85   85 08      112         STA 08
4F87   A9 BE      113         LDA #$BE
4F89   85 07      114         STA 07
4F8B   85 09      115         STA 09
4F8D   A9 00      116         LDA #00
4F8F   8D E1 BF   117         STA $BFE1
4F92   60         118         RTS
4F93              119 ;
4F93              120 ;******* C'EST PARTI !!!! ******
4F93              121 ;
4F93              122         ORG $5000
5000              123 CTRL    EQU $4F00               ; MAIN LOOP
```

```
5000              124 C/GX    EQU $4F01
5000              125 C/DX    EQU $4F02
5000              126 C/B     EQU $4F03
5000              127 C/H     EQU $4F04
5000              128 TOUR    EQU $4F05
5000              129 H/L     EQU $4F06
5000              130 D/L     EQU $4F07
5000              131 ETAGE   EQU $4F08
5000              132 SWAP    EQU $4F09
5000              133 PAIR    EQU $4F0A
5000              134 RD      EQU $4F0B
5000              135 RD2     EQU $4F0C
5000              136 LIG     EQU $4F0D
5000              137 COL     EQU $4F0E
5000              138 TIME    EQU $4F0F
5000              139 MONSTRE EQU $4F10
5000              140 RESL    EQU $4F11
5000              141 RESH    EQU $4F12
5000              142 X       EQU $4F13
5000              143 Y       EQU $4F14
5000              144 L       EQU $4F15
5000              145 H       EQU $4F16
5000              146 XFLAG   EQU $4F17
5000              147 XM      EQU $4F18
5000              148 YM      EQU $4F19
5000              149 NPAS    EQU $4F1A
5000              150 NPAST   EQU $4F1B
5000              151 NCPAST  EQU $4F1C
```

```
552F              805 ;****************************
552F              806 ;
552F              807 ;    FIN DE LA BOUCLE
552F              808 ;
552F              809 ;****************************
552F   20 5F 54   810 SUITE   JSR SSN
5532   AD E1 BF   811         LDA $BFE1
5535   C9 01      812         CMP #1
5537   D0 01      813         BNE >1
5539   60         814         RTS
553A   C9 02      815 >1      CMP #2
553C   D0 01      816         BNE >1
553E   60         817         RTS
553F   20 A5 53   818 >1      JSR REDEF
5542   A2 00      819         LDX #$0                ;FIN DE LOOP
5544   A0 00      820 >2      LDY #$0
5546   EA         821 >1      NOP
5547   EA         822         NOP
5548   EA         823         NOP
5549   C8         824         INY
554A   D0 FA      825         BNE <1
554C   E8         826         INX
554D   E0 13      827         CPX #$13
554F   D0 F3      828         BNE <2
5551   AD 08 02   829         LDA $208
5554   C9 A9      830         CMP #$A9
5556   D0 01      831         BNE >1
5558   60         832         RTS
5559   4C 00 50   833 >1      JMP DEBUT

FIN DE LA PASSE 2

               FIN= $555C
```

PETITES ANNONCES

La place réservée aux petites annonces est limitée. En conséquence, celles-ci passent dans leur ordre d'arrivée. Par contre, les petites annonces farfelues sont systématiquement rejetées... De même, comme il est peu probable qu'il existe des "généreux donateurs" de matériels ou de logiciels, les petites annonces faisant appel à ces philantropes ne seront insérées que si la place libre le permet.
Seront refusées toutes les annonces visant à vendre ou échanger des listes ou copies de logiciels non garantis "d'origine", ainsi que toute annonce incitant au piratage.
En conséquence, réfléchissez bien avant d'envoyer vos textes.
Les petites annonces doivent impérativement nous parvenir sur la grille (découpée ou photocopiée), le texte étant rédigé à raison d'un caractère par case.
Enfin, toute annonce non accompagnée de timbres ne sera pas insérée.

Cause abandon activité, vends combinaison planche à voile absolument neuve : 250 F. D. BONOMO, tél.: 99.52.98.11.

Vends ATMOS + moniteur + magnéto + logiciels + meuble + livres + carte Eureka + joystick : 2500 F. Inter CGV Péri-sec. : 300 F. Tél.: 30.35.83.26.

Cherche imp. MCP 40 hors usage remb. frais et donne K7 Xenon 3 en échange. Tél.: 64.21.34.90 le soir, Gérard. Merci.

Vends Microdisc et Sedoric sous garantie (21.01.86) état neuf + manuel. Prix à débattre. Tél.: 69.21.29.66 après 19 h.

Vends ORIC-1 : 300 F. "Speak Easy" synthé + prog 27 k : 400 F. Logiciels 1/2 prix + port. Tél.: 85.84.93.16.

Vends modem Digitelec pour ORIC ATMOS avec logiciel : 1000 F. Tél.: 48.54.22.35 demander Michel.

Vends ORIC-ATMOS + 6 livres + 18 cassettes (Aigle d'Or, Ile Maudite). Le tout acheté en juin 1985 : 2000 F.

Cherche vendeur d'une interface prog. joystick. Offre jusqu'à 425 F. Tél.: 39.74.76.33 après 17 h.

Vends progamme d'accentuation française pour écran et imprimantes du logiciel AUTHOR. 100 % langage machine : 80 F comprenant : cassette, notice, lettres adhésives. Notice seule contre deux timbres à 2,20 F. J.P. SUAU, 13 square Protis, 13002 MARSEILLE.

Possesseur ATMOS recherche THEORIC 1, 2, 3, 5 et 7. Echange chaque THEORIC contre 4 jeux. Ecrivez-moi pour me dire les numéros que vous avez ; je vous renverrai la liste. Gilles LAUX, 3 allée des Coquelicots, 57310 GUENANGE.

Vends ultra-images Arsène Lapin - Drivers - tous. Adapt. N° 1 Styx, bon état, le tout : 350 F. Tél.: 89.58.74.40 (HR).

Vends ou échange adapt. prog. commerce ORIC-ATMOS adapté Jasmin-2. Hervé MIGNOT, tél.: (1) 30.64.51.16.

Vends ORIC-1 ROM ORIC + ROM ATMOS + microdisc + 50 jeux et utilitaires sur K7 et disquette : 2800 F, cause achat Téléstrat. Tél.: 20.94.03.65.

Vends Vortex (traitement de texte) : 140 F. Formule 1 : 80 F. Hyperspace 4 : 60 F pour ORIC-ATMOS. Tél.: 77.23.37.45.

Vends ATMOS 48 k + interrupteur + matériel péritel + jeux + alim. 9 V neuve + carte manette : 1800 F. Tél.: (1) 46.21.33.12 après 19h30.

Vends imprimante GP 100 - Mark II (sans cordon) : 1000 F. M. DEPRET, 10 rue des Martyrs de Soweto, 51000 CHALONS/MARNE.

Vends ORIC-ATMOS + lecteur + adapt. de prise péritel + 200 jeux + livres : 2000 F. Tél.: 30.34.15.16 après 18h.

ORIC-ATMOS cherche contacts pour échanges sur Toulouse. Patric FROMENT, 1 rue Cordeau Dany, 31200 TOULOUSE, tél.: 61.47.49.37.

Vends ORIC-1 48 k (12/84) + alim. + péritel + livres + 10 K7 + synthé : 2850 F. Jim BLEY, Le Flore B1, Av. Martyrs Résistance, 06600 ANTIBES.

Vends GP 100A + Supercopy écran + Vortext (trait. texte) + papier : 1600 F. Tél.: 87.96.02.26 le soir.

Vends ATMOS peu servi + lecteur K7 + 12 livres + K7. Valeur 4300 F (factures), vendu 2000 F. M. GAILLARDOT, 37290 YZEURES SUR CREUSE, tél.: 47.94.55.41.

Vends Jasmin 2 avec assembleur, Easytext, Tableur Multifich, Livre du TDOS, 10 disquettes vierges : 3000 F. Tél.: 48.20.61.30 M. DURVILLE.

Poss. ATMOS cherche contact région Le Mans pour Synth. vocal (Micro Système) MEA 8000. Tél.: 43.25.62.99 après 19h.

ATMOS et JASMIN cherche logiciels. Claude LEGRAND, 3 place du marché, 77400 LAGNY, tél.: 64.30.12.63.

Achète revues THEORIC n° 5 et 13, reuve MICR'ORIC n° 2, revues CPC 1, 2, 3, revue ORIC OWNER, 1, 2, 3. M. Gauthier Sublett, tél.: 57.74.69.56.

ATMOS cherche correspondant(e) pour échanges prog. et idées Rhône Alpes. Tél.: 74.03.84.80.

STOP ! Améliorez vos prog. jeux ou anim. graph. grâce à système DAO. C. GAYRAUD, 32 av. de Villandry, 44470 CARQUEFOU, tél.: 40.50.83.83.

Echange EASYTEXT et/ou Jasmincalc contre Multifich, ForthH, Assembleur, Origraph ou autre logiciel Jasmin. M. GOBART, tél.: 64.34.28.68.

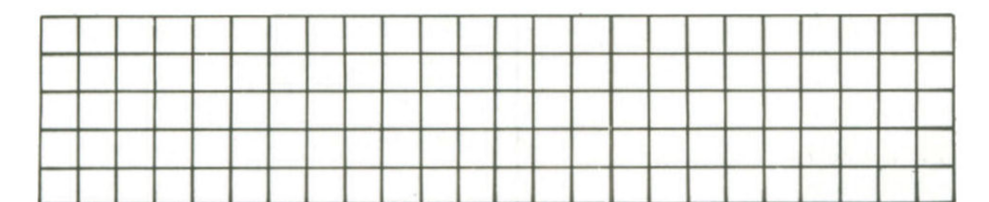
Possess. ATMOS cherche contacts pour échanges idées et logiciels. A. RUEL, La Suchère, 43400 CHAMBON SUR LIGNON.

Vends ATMOS + 60 prog. + interface manette + manette + Easytext + 5 disquettes Jasmin, le tout ou séparément. Tél.: 79.32.50.36 le soir.

Urgent, vends ATMOS (8/84, tbe) + câbles + livres + revues (THEORIC) + moniteur couleur 16" (tout neuf) + jeux. Prix à débattre, tél.: 6373070

ANNONCEZ-VOUS!

les petites annonces et les messages



T 20

Coupon à renvoyer accompagné de 3 timbres à 2,20 F à :
SORACOM, La Haie de Pan, 35170 BRUZ

